IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received March 27, 2020, accepted April 18, 2020, date of publication April 22, 2020, date of current version May 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2989586

Scheduling Dynamic Multicast Requests in Advance Reservation Environment for Enterprise Video Conferencing Systems

ZHIWEN LIAO AND LING ZHANG, (Associate Member, IEEE)
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 video conferencing system, the enterprise network can be shared by hundreds of video conferences. On the other hand, participants may not arrive at the same time and stay until the end of the conference. The abilities to reserve resources in advance, as well as effective dynamic multicast when participants can join and leave the conference at any time, are essential in the distributed multi-party video conferencing systems. However, the effective advance reservation strategies of the dynamic multicast requests for a heavy traffic case still remains open. In this paper, we investigate the problem of Maximizing the number of admitted Dynamic Multicast requests in the Advance Reservation environment (MDMAR) for the enterprise video conferencing system. We take two path schemes of a fixed path and variable paths, as well as a heterogeneous bandwidth reservation model into account. Firstly, we prove that the MDMAR problem is NP-complete and formulate it mathematically as an integer linear program (ILP) for small networks. Then, we develop greedy algorithms and simulated annealing (SA) algorithms for enterprise networks. Comparative simulations are performed to evaluate the heuristic algorithms for both small networks and enterprise networks. We find that the SA algorithms can provide within 6% lower optimal solutions than the ILP algorithms for our small network, and up to 10% improvement over the greedy algorithms for the large campus or enterprise network.

INDEX TERMS Enterprise video conferencing system, advance reservation (AR), integer linear program (ILP), dynamic multicast.

I. INTRODUCTION

The statistics from Global Workplace Analytics and Flexjobs show that regular telecommuting grew 115% in the past decade, nearly 10 times faster than the rest of the workforce [1]. According to FlexJob's 2018 annual survey of more than 3,000 respondents, 97% said a flexible or telecommuting job would have a huge improvement or positive impact on their overall quality of life [2]. Recently, 36Kr reported that, in China, nearly two hundred million people opened "cloud office" on February 3rd in 2020 [3]. As we enter a new decade, we can forecast video conferencing is getting more attention than ever before. As we know, video conferencing consumes substantial network bandwidth and requires an ultra-low end-to-end latency, which is 150ms for one-way latency [4]. Such bandwidth-hungry applications have driven the Internet IP video traffic grow exponentially. Globally, IP video traffic will account for 82% of traffic by 2022 [5]. Moreover, with the increasing video resolution, more bandwidth will be required. For example, the bit rate for video Ultra-High-Definition (UHD, or 4K) is about 15 to 18Mbps [5]. However, the network bandwidth is limited. Therefore, it is crucial to discover an effective way to manage the network resources and provide the Quality-of-Service (QoS) guarantees.

Centralized conference (XCON) Working Group has proposed various conferencing scenarios, one of which is called the reserved conference [6]. The resource reservation for this

The associate editor coordinating the review of this manuscript and approving it for publication was Peng-Yong Kong.

type of conference is typically done by an out-of-band mechanism in advance of the actual conference time [6]. Based on the available resources in the future and the conference information, advance reservation (AR) [7] can improve the resource utilization as well as QoS.

The resource reservation protocols (RSVP) [8] and the enhancements of RSVP to cope with the advance resource reservation have been proposed to offer the QoS guarantees in 1990s [9]. However, due to the delay in communication protocols and the distributed management of routers, RSVP is not widely used for video transmission in the IP network. By contrast, more works related to AR for the optical network have been proposed in the past two decades [7], [10], [11]. This is mainly because the network is under centralized control by a central controller, which is responsible for routing and wavelength assignment (RWA) and establishing lightpaths for all connection requests on behalf of all network nodes [10]. Typically, there are two types of network traffic: Static and dynamic [10]. The static AR requests are known ahead of time while the dynamic AR requests arrive according to some stochastic process. The authors in [10] stated that the problem for the static AR requests is to establish as many lightpaths for requests as possible for a given number of wavelengths while the problem for the dynamic AR requests is to minimize the blocking probability. In this paper, we mainly address the static traffic model with the goal to maximize the number of admitted AR requests as possible for the limited network resources and the QoS level.

In a video conferencing group, the same video can be simultaneously transmitted to multiple participants. People have demonstrated that using multicast in place of multiple unicast streams improves the network resource utilization [12]. Depending on the behavior of the multicast group participants, a multicast group can be either 1) Static. All the participants join the session at the beginning and leave until the session ends. 2) Dynamic. The participants join and leave the session at any time independently [13]. The static multicast problem is also known as the Steiner tree problem, which is known to be NP-complete. Waxman *et al* [14] mainly studied the unitary dynamic multicast problem only for a unitary session. However, in a video conference, every participant can send and receive the video stream, and the participants may not arrive at the same time and stay until the end of the conference. Consequently, reserving bandwidth for the static multicast may result in a waste of resources. In this paper, in order to improve the resource utilization, we reserve the bandwidth for every video stream connection, and the users are asked not only the bandwidth but also the start time and end time for every connection in the conference.

While there are a few works dealing with the static and unitary dynamic multicast advance reservation problem for the IP network [13], [15], [16], little attention has been paid for the dynamic multicast advance reservation. Moreover, we find that these work only consider the immediate advance reservation, where the session starts when the reservation is done, for the unitary multicast session. Like RSVP, IP multicast-based video transmission is also not widely applied in the current IP networks. In particular, when the members in the multicast group enter or leave at any time, the routers do not have the capability to make adjustment dynamically and adaptively. Therefore, it's not surprising that there are few works on multicast advance reservation.

Recently, software-defined networking (SDN) has exhibited effective performance in cloud network [17]. It decouples the control plane from the data plane and leverages centralized network control and management. OpenFlow [18] is a standard communication protocol that enables SDN. It is convenient to implement multicast on top of the SDN controller instead of deploying any distributed multicast routing protocols. In this paper, we deploy a multicast AR system on top of the SDN controller. Moreover, because the network we study is the enterprise network, it is feasible and practical to deploy resource management strategies for multicast.

In this paper, we consider a general scenario for video conferencing systems where there are hundreds of conferences online at the same and the participants are allowed to enter the conference at different times and leave before the end of the conference. We investigate the problem of Maximize the number of admitted Dynamic Multicast requests in the Advance Reservation environment (MDMAR) for the enterprise video conferencing systems. By combining dynamic multicast and static advance reservation, every participant is asked to declare the required bandwidth, as well as the joining and holding time. Due to the various types of video terminals, which have different receiving capacities, the required bandwidth may be different for different receivers. In this paper, we consider a heterogeneous resource reservation model. When the participant joins the existing multicast tree, we don't reconstruct the existing multicast tree.

Four types of AR scheduling problems were formulated in [19] to exhaustively combine different path schemes and bandwidth constraints. Different from the large file transferring in [19], we think the required bandwidth of the real-time video streaming remains unchanged within the session time. In this paper, we address the fixed path and variable path schemes for MDMAR problem. Time domain [7] is an important aspect in AR. As the requests increase, the time domain can be divided into a series of fine-grained time intervals, which results in more complexity to manage these time-variable available bandwidth for the time domain. Timeslot-based approach [20] is verified effectiveness for the management of the time domain. In this paper, we introduce the dynamic timeslot-based approach to manage the time domain. The main contributions are summarized as follows.

1) We consider a general and practical situation where there are a large number of conferences online at the same time, and the participants may not arrive at the same time and leave until the end of the conferences. To the best of our knowledge, this is the first time to reserve bandwidth for the dynamic multicast requests for a heavy traffic load.

2) We prove that the MDMAR is NP-Complete. A mathematical model of the problem is formulated, taking the path constraints, the entity of the conference and the dynamic timeslots into account for small networks.
3) Four heuristics for the fixed path and variable path are proposed to solve the problem for large networks.

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 the ILP model are discussed in Section IV. Section V presents the four heuristics. Simulations and evaluations are shown in Section VI. Finally, Section VII summarizes this paper.

II. RELATED WORK

Recently, there have been significant works for video conferencing systems based on multicast deployed on top of the SDN controller [21]–[23]. However, they did not consider the dynamic scenario as well as the advance reservation.

Advance reservation is an effective way for the large file data transfer and real-time streams. There are significant works on unicast advance reservation. The authors in [24] formulated an ILP model for static advance reservation requests of file transfer with the objective of maximizing the number of admitted requests. The authors in [25] also proposed an ILP model for the advance reservation requests on top of SDN with the goal to achieve optimal resource utilization. AR can also provide differentiated services to users to increase the revenue of the network. For example, in [26], [27], the authors focused on a revenue-driven dynamic resource provisioning approach to increase the profit of the network. In [28], the authors proposed a fine-granularity QoS micropayment system that allows users to prepay for guaranteed bandwidth reservation for a period of time. Different from those works, which are for the unicast advance reservation requests, we address the multicast advance reservation problem whose ILP model is greatly different. Moreover, we take two types of path schemes and dynamic timeslots into account, which makes the model more complex.

There are a few works related to multicast reservation. The work in [29] presented a general framework of admission control and resource reservation for multicast sessions and developed algorithms aiming to efficiently utilize network resources. However, we find that the multicast resource reservation has not greatly developed in the past two decades for the IP network. This is mainly because of the complex protocol control, e.g, Internet Group Management Protocol(IGMP) and RSVP, as well as the distributed network view. With the appearance of SDN and Network Function Virtualization (NFV), which provide a centralized view of the network, there are a few works related to the multicast advance reservation based on SDN and NFV. The authors in [30], [31] focused on the balance between guaranteed-bandwidth multicast traffic and best-effort traffic and implemented the scalable multicast routing algorithm based on the NFV. The work in [32] addresses the dynamic traffic engineering and the rerouting overheads of the dynamic multicast tree. However, those works related to resource reservation only consider the immediate reservation, where the data transmission starts upon the request is admitted. The immediate reservation is different from the advance reservation in this paper.

Different from the IP network, there are significant works related to multicast advance reservation on the optical WDM network. The authors in [11], [33], [34] formulate ILP models and design heuristics for the static multicast routing and spectrum assignment in advance reservation environment. Though the ILP formulation for optical multicast advance reservation show reference for our work, the model for the IP network still has a great difference from that for the non-optical network. The authors in [35] investigated the multicast formulation problem with the objective of minimum cost subjected to QoS constraints. Instead of building the optimal tree for the problem, the authors formulated the hierarchies for the problem. Though the cost of hierarchies is better than the tree, it results in greatly increasing execution time because it is derived from the multi-graph obtained by duplicating each topology $|D|$ times, where D is denoted as the set of destinations in a multicast session.

Group multicast is one type of multicast in which each member node from the group may multicast data to all other members from the same group, i.e, each member node being both an information source and destination [36]. A typical example is that for the discussion video conference every participant can send video streams to, as well as receive video streams from, the other participants in the same conferencing group. It is not difficult to observe that the group multicast will demand higher bandwidth resources than the corresponding "single source" multicast routing. Similar to multicast, group multicast can be either static or dynamic. In dynamic group multicast, participants may join or leave the multicast group during the lifetime of the multicast connection [37]. Most of the exiting works related to group multicast address the minimum cost of the unitary group while little attention on the multiple group multicast or on the advance reservation.

III. AR-ENABLED NETWORK ARCHITECTURE

In our previous work [38], we have designed a video conferencing system (VCS). The VCS adopts a centralized signal control and a distributed media mixing mode, which is similar to architecture proposed in the RFC 5239 [39]. Then, we extended the VCS by adding the advance reservation system in [40]. However, this system only focused on the case of resource reservation in advance, and does not consider the case of no resource reservation. In fact, both of them exist in the video conference system, and we can't ignore the case in which conferences don't need to reserve resources. This paper will provide a global view of how they work together in a VCS.

Fig.1 shows the AR-enabled network architecture and components of a VCS. The architecture consists of four planes: the application plane, the network orchestration plane,

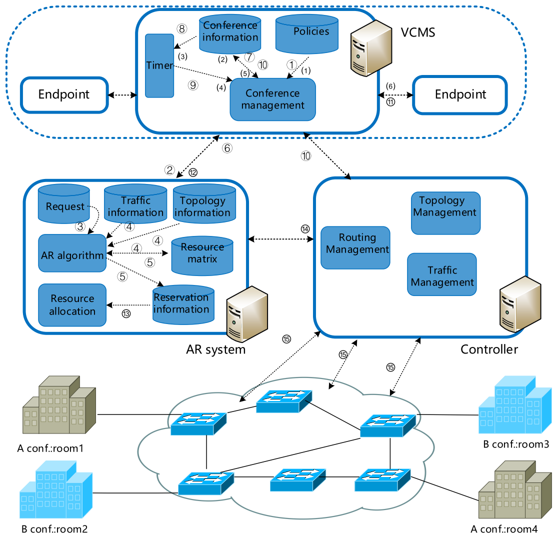


FIGURE 1. **AR-enabled network architecture and components of a VCS.**

the control plane and data forwarding plane. The application plane is a VCS, which consists of a video conferencing management system (VCMS) and endpoints. The VCMS is the center of conference management and conference control. Endpoint is a user agent that supports media mixing and can send or receive media. The AR system is in the network orchestration plane, which provides the differential service to the application plane and communicates with the control plane using the northbound API. The SDN controller is in the controller plane, which is responsible for the management and control of the network. In the data plane, the participants in geographically distributed conference rooms are connected to the shared enterprise networks through interconnected switches.

Since conferences are centrally managed, all conferences must be registered in VCMS first. When registering, users are required to provide information, including start time, end time and terminals. If users want to guarantee QoS of every video stream in the conference, they also need to provide the information of each video stream, including source address, destination address and required bandwidth.

If a conference needs resource reservation in advance, it is first executed admit control based on the policies (step①). If it is permitted to reserve resource, it is delivered to the AR system (step②). At the AR system, it is stored in the request base, waiting to scheduled periodically (step③). Once scheduled, it is processed by the AR algorithm based on the resource matrix, which preserves the available bandwidth information in the future generated based on the request, topology information and traffic information (step④). If the conference can be accommodated with the required bandwidth, the AR system stores the reservation information, and updates the resource matrix (step⑤). Then, the AR system returns the admitted information to the VCMS (step⑥). The VCMS saves the conference in the conference information base (step⑦). A timer in the system monitors the conference information base periodically (step⑧). Once the start time of the conference is due, the timer notifies the conference management module (step⑨). The system reads the conference information from the base and set the conference alive (step⑩). Only when the conference is alive can endpoints join the conference. Then, an endpoint enters the conference and sends a video stream connection request (step⑪). The request is then delivered to AR system (step⑫). The resource allocation module reads the corresponding reservation information from the reservation information base (step⑬), and delivers it to the controller (step⑭). The controller sends the resource reservation messages to the switches on the path (step⑮). Finally, the endpoint receives the video streaming on the assigned path.

Although a conference does not demand resource reservation, it is also performed admit control based on the policies (step(1). If it is admitted, it is firstly stored in the conference information base (step(2)). Then, the timer monitors the base periodically (step(3)). Once the start time of the conference is due, the timer notifies the conference management module (step(4)). The conference management module reads the conference and sets it alive (step(5)). Next, endpoints enter the conference (step(6)) and exchange the video streams with the other members.

IV. NETWORK MODEL AND PROBLEM DESCRIPTION

In this section, we discuss the theoretical model for the MDMAR problem and prove the problem is NP-Complete.

A. NETWORK MODEL AND PROBLEM DESCRIPTION

We define the topology of the 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 investigate the static advance reservation case where all the AR requests are known ahead of time. For every participant, he is allowed to join and leave the conference at any time independently. By combing the advance reservation and dynamic multicast, every participant is asked to provide the start time, holding time and the required bandwidth. Our goal is to maximize the number of the admitted requests. The video stream connection requests of one conference must be admitted in the entity. Due to different terminal capacities, the required bandwidth of each participant may be different for different receivers. Therefore, we consider a heterogeneous reservation model. Moreover, no reconfiguration is allowed as a new participant enters the multicast tree.

A user makes a conference request, which contains multiple video stream connection requests with time attribute. The set of all conference requests is denoted as $C = \{c_1, c_2, \ldots, c_m, \ldots\}$. All video stream connection requests are stored in R, which is denoted as $R = \{r_1, r_2, \ldots, r_k, \ldots\}$. Each connection request is represented as

$r_k = (c_k, s_k, d_k, st_k, et_k, b_k)$, where k is the index in R, c_k is the conference id that r_k belongs to, s_k is the source of the connection request, d_k is the destination, st_k is the start time of the connection request, et_k is the end time of the connection request, b_k is the required bandwidth. When a video stream connection request $r_k = (c_k, s_k, d_k, st_k, et_k, b_k)$ arrives, if it is the first connection request starting from s_k, we builds a multicast tree that starts from s_k, and reaches d_k with the reserved bandwidth b_k in the time span of (st_k, et_k); otherwise, we firstly find the multicast tree that is available in the time span (st_k, et_k) and starts from s_k in the same conference c_k. Then, we add the receiver d_k to the multicast tree in the shortest path with the sufficient bandwidth in the time span of (st_k, et_k) from the source node s_k. The MDMAR problem is defined as follows.

Definition **MDMAR** (G, R): Given a network $G = (V, E, B)$ and a conference request set $C = \{c_1, c_2, ..., c_m, ...\}$. Each conference request contains multiple video stream connection requests. All the video stream requests are stored in the set, $R = \{r_1, r_2, ..., r_k, ...\}$. A multicast tree is built for the connection requests with the same source node of a conference, and each connection is accommodated with sufficient bandwidth within the corresponding time span. The video stream connection requests of a conference are admitted in the entity. Our goal is to maximize the number of admitted requests in the set R without violating the link bandwidth capacity.

B. NP-COMPLETENESS PROOF

In this section, we show the NP-completeness of MDMAR. Firstly, we consider a simple version of MDMAR where 1) there are only two nodes, s and d, in the network. Therefore, there is only one path (s, d) in the network, and the bandwidth of link (s, d) is assumed as b. 2) each conference only has one video stream connection request starting from node s and reaching d, which is denoted as $r_k = (c_k, s, d, st_k, et_k, b_k)$. There are multiple video streaming connection requests, which are stored in the set $R = \{r_1, r_2, ..., r_k, ...\}$. 3) the time span that users can hold conferences is from T_s to T_e. This simple case is referred to as SIMPLE-MDMAR. We illustrate the simple case in Fig. 2.

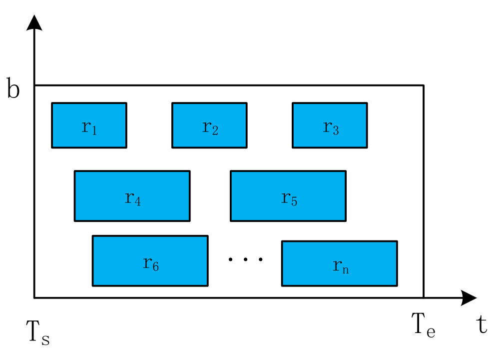


FIGURE 2. Requests in the SIMPLE-MDMAR problem.

Lemma 1: SIMPLE-MDMAR is NP-complete.

proof: As illustrated in Fig. 2, we view the initial bandwidth b within the time span of (T_s, T_e) as a bin and the video stream connection requests as items, and the objective is to maximize the number of packed items, or to minimize the total area of the admitted items. This way, SIMPLE-DMARM can be viewed as the two-dimensional rectangle bin packing problem, which has been proved to be NP-complete [41].

Theorem 1: MDMAR is NP-complete.

proof: In SIMPLE-MDMAR, we consider a simple network in which there is only one path. In MDMAR, the network is a more general and more complex network. As a simple version is NP-complete according to Lemma 1, so is the more complex problem according to the principle of proof by restriction [42].

The time domain management is very important for advance reservation. Different time domain management approaches can result in different computational complexity and acceptance ratio. As the dynamic timeslot-based resource management approach shows effectiveness for a heavy traffic load, we use this approach to solve the time processing and manage the resources. In this paper, both the ILP model and heuristics are based on the dynamic timeslot-based resource management approach.

C. THE DYNAMIC TIMESLOT-BASED RESOURCE MANAGEMENT APPROACH

Fig. 3 shows the way of time processing of a request. The horizontal coordinate in Fig. 3 represents the time domain (i.e., the initial largest timeslot), and the vertical coordinate represents the consumption and remaining bandwidth of a link. In the dynamic timeslot-based approach, the start time and end time are respectively rounded down and up to the nearest integer times of the granularity. The granularity is the smallest timeslot that the time can be partitioned into, i.e., the timeslot between the two adjacent dashed lines, as shown in Fig. 3. The length of the time domain is set as d. We use an array T to store the ordered discrete time points on the horizontal coordinate. Initially, $T = [0, d]$ and there are two timeslots $\{(0, 0), (0, d)\}$. As r_1 arrives, the start time st_1 is rounded down to T_1, and et_1 is rounded up to T_2, as shown in Fig. 3. Then, $T = [0, T_1, T_2, d]$ and there are four timeslots: $\{(0, 0), (0, T_1), (T_1, T_2), (T_2, d)\}$.

Fig. 4 shows the dynamic partition process of timeslots when requests arrive one by one. As r_2 arrives, the requested times are firstly processed as in the Fig. 3. Then, the processing times are insert sort into the array T, as shown in Fig. 4. At this time, $T = [0, T_1, T_2, T_3, T_4, d]$ and there are six timeslots: $\{(0, 0), (0, T_1), (T_1, T_2), (T_2, T_3), (T_3, T_4), (T_4, d)\}$. The *ith* timeslot is denoted as (T_{i-1}, T_i) while $i \geq 1$. When $i == 0$, the timeslot is denoted as (T_0, T_0), that is, $(0, 0)$.

For the dynamic timeslot-based resource management approach, each timeslot accumulates the available bandwidth of all the links in the network, and the available bandwidth of each link remains unchanged in the timeslot. Here,

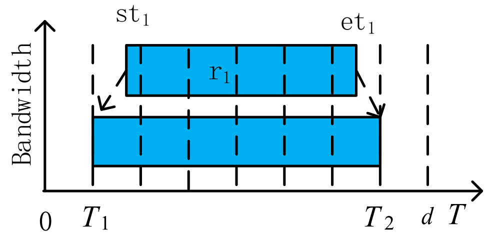


FIGURE 3. The way of time precessing of a request.

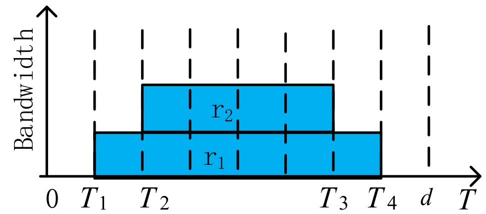


FIGURE 4. Dynamic partition method of timeslot for the dynamic timeslot-based approach.

we consider two path schemes for a video stream connection: a fixed path for the whole session time and various paths for the whole session time. The fixed path scheme calculates a a fixed path for the request, and the video is transmitted on this path for the whole session time while the various path scheme calculates a path in each timeslot, and the video is transmitted on various paths for the whole session time.

In the following, we design an ILP model and heuristics for the two path schemes based on the dynamic timeslot resource management approach respectively.

D. ILP MODEL

In this section, we formulate the MDMAR problem mathematically as an ILP. We use an ILP solver (LINGO) to find an optimal solution. Since the MDMAR problem is NP-complete, solving an ILP is also NP-complete. When we consider a heterogeneous reservation model, the MDMAR problem will be formulated as an integer non-linear programming (INLP). This will greatly increase the computational overhead. In this section, we assume that the receivers in the same session require the same bandwidth for simplicity. The parameters of the ILP are as shown below.

1) PARAMETERS

- $b(u,v) \in B$: the initial bandwidth capacity of link (u,v);
- C: the set of conference requests;
- R: the set of video stream connection requests;
- $r_k = (c_k, s_k, d_k, st_k, et_k, b_k)$: the request r_k;
- T: the array that stores ordered time points of time domain, and the initial time is denoted as is T_0;
- $b_{(u,v,t)}$: the initial bandwidth capacity of link (u,v) in the timeslot (T_{t-1}, T_t);
- S_m: the set of all source nodes of the conference C_m.
- g: the size of granularity for the dynamic timeslot.

2) VARIABLES

- f_m: Boolean variable that equals 1 if the conference c_m is admitted, otherwise $f_m = 0$;
- x_k: Boolean variable that equals 1 if the request r_k is admitted, otherwise $x_k = 0$;
- $y_{k,t}^{(u,v)}$: Boolean variable that equals 1 if link (u,v) is used to deliver the request r_k in the timeslot (T_{t-1}, T_t), otherwise $y_{k,t}^{(u,v)} = 0$.
- $z_{m,v,t}^{(u,v)}$: Boolean variable that equals 1 if the multicast tree for the conference c_m sourced from $v \in S_m$ flows by link (u,v) in the timeslot (T_{t-1}, T_t), otherwise $z_{m,v,t}^{(u,v)} = 0$;
- $h_{m,v,t}^{(u,v)}$: the reserved bandwidth of link (u,v) by which the multicast tree for the conference c_m sourced from v flows in the timeslot (T_{t-1}, T_t). If link (u,v) is not used to deliver the tree, $h_{m,v,t}^{(u,v)} = 0$;
- $x_{k,t,u}$: Boolean variable that equals 1 if the node u is used to deliver the request r_k in the timeslot (T_{t-1}, T_t), otherwise $x_{k,p,u} = 0$;
- $x_{k,t}$: Boolean variable that equals 1 if the request r_k is admitted in in the timeslot (T_{t-1}, T_t), otherwise $x_{k,t} = 0$.

3) OBJECTIVE

The objective function, shown in (1), maximizes the number of admitted requests, as well as trying to minimize the reserved bandwidth for all the conferences for the whole time domain. Since the two objectives are contradictory, we add a negative sign to the number of admitted requests, as shown in the first part of (1), and find the minimum value for (1). The second part is to minimize the reserved bandwidth and normalized to ensure it is lower than 1 so that it will not interface with the first primary objective.

$$\min(-\sum_{r_k \in R} x_k + \frac{\sum_{c_m \in C} \sum_{V \in S_m} \sum_{(u,v) \in E} \sum_{t \in T} h_{m,v,t}^{(u,v)}}{\sum_{(u,v \in E} \sum_{t \in T} b_{(u,v,t)}}). \tag{1}$$

4) SUBJECT TO

1) Flow Conservation Constraints:

$$\sum_{(v,u) \in E} y_{k,t}^{(v,u)} = 0, \quad u = s_k, \quad \lfloor st_k/g \rfloor < t \le \lceil et_k/g \rceil, \ \forall r_k \in R. \tag{2}$$

$$\sum_{(u,v) \in E} y_{k,t}^{(u,v)} = x_{k,t,u}, \quad u = s_k, \quad \lfloor st_k/g \rfloor < t \le \lceil et_k/g \rceil, \ \forall r_k \in R. \tag{3}$$

$$\sum_{(u,v) \in E} y_{k,t}^{(u,v)} = 0, \quad u = d_k, \quad \lfloor st_k/g \rfloor < t \le \lceil et_k/g \rceil, \ \forall r_k \in R. \tag{4}$$

$$\sum_{(v,u) \in E} y_{k,t}^{(v,u)} = x_{k,t,u}, \quad u = d_k, \quad \lfloor st_k/g \rfloor < t \le \lceil et_k/g \rceil, \ \forall r_k \in R. \tag{5}$$

$$\sum_{(v,u) \in E} y_{k,t}^{(v,u)} = \sum_{(u,v) \in E} y_{k,t}^{(u,v)}, \quad u \neq s_k \wedge u \neq d_k, \quad \lfloor st_k/g \rfloor < t \le \lceil et_k/g \rceil, \ \forall r_k \in R. \tag{6}$$

$$\sum_{(v,u)\in E} y_{k,t}^{(v,u)} = x_{k,t,u}, \quad u \neq s_k \wedge u \neq d_k, \quad \lfloor st_k/g \rfloor < t \leq \lceil et_k/g \rceil, \ \forall r_k \in R. \quad (7)$$

$$x_{k,t,u} = x_{k,t}\ u = s_k \vee u = d_k, \quad \lfloor st_k/g \rfloor < t \leq \lceil et_k/g \rceil, \ \forall r_k \in R. \quad (8)$$

(2) and (4) force the source node and the destination node to have no incoming links and outgoing links respectively for any request $r_k \in R$. (3) and (5) force the source node and the destination node to have only one outgoing link and one incoming link respectively, and (8) adds the source node and destination to the corresponding multicast tree. (6)-(7) ensure that the intermediate nodes of the multicast tree have the same number of incoming links and outgoing links and the number is equal to at most 1. (2)-(8) complete the general constraints for constructing a path for a video stream connection request.

2) Traffic Aggregation Constraints:

$$z_{m,v,t}^{(u,v)} \geq y_{k,t}^{(u,v)}, \quad \{k : c_k = m, s_k = v\}, \ \forall c_m \in C, \quad \lfloor st_k/g \rfloor < t \leq \lceil et_k/g \rceil, \ \forall (u,v) \in E. \quad (9)$$

$$z_{m,v,t}^{(u,v)} \leq \sum_{\{k:c_k=m,s_k=v\}} y_{k,t}^{(u,v)}, \ \forall c_m \in C, \quad \lfloor st_k/g \rfloor < t \leq \lceil et_k/g \rceil, \ \forall (u,v) \in E. \quad (10)$$

(9)-(10) ensure that the traffics sent from the same source node to different receivers are aggregated since we use a multicast tree to serve all the receivers that require the same video streams.

3) Fixed Path Scheme:

$$y_{k,t}^{(u,v)} = y_{k,\lfloor st_k/g \rfloor+1}^{(u,v)}, \quad \lfloor st_k/g \rfloor + 1 < t \leq \lceil et_k/g \rceil, \quad \forall r_k \in R, \ \forall (u,v) \in E \quad (11)$$

In this paper, we consider two path schemes: a fixed path through the whole session time and variable paths for different timeslots. (11) is the constraint for the fixed path scheme while no this constraint is for the variable path scheme.

4) Admitted in entity:

$$x_{k,t} = x_k, \quad \lfloor st_k/g \rfloor < t \leq \lceil et_k/g \rceil, \ \forall r_k \in R. \quad (12)$$

$$x_k = f_m, \quad \{k : c_k = m\}, \ \forall c_m \in C. \quad (13)$$

(12) ensures that the video stream connection request r_k is admitted only when the request is admitted for all the timeslots, i.e, the whole session time, and (13) ensures that the conference c_m is admitted only when all the video stream connection requests are all admitted.

5) Bandwidth Constraints:

$$h_{m,v,t}^{(u,v)} = z_{m,v,t}^{(u,v)} \times b_k, \ \{k : c_k = m, s_k = v\}, \ \forall c_m \in C, \quad \lfloor st_k/g \rfloor < t \leq \lceil et_k/g \rceil, \ \forall (u,v) \in E. \quad (14)$$

$$\sum_{\forall c_m \in C} \sum_{\forall v \in S_m} h_{m,v,t}^{(u,v)} \leq b_{(u,v,t)}, \quad \forall t \in [1, |T|-1], \ \forall (u,v) \in E. \quad (15)$$

(14) calculates the reserved bandwidth on each link (u, v). Note that the requests that have the same source node require the same bandwidth, that is, the value of b_k for the same source node is equal. (15) ensures that the reserved bandwidth for all the requests cannot be more than the link capacity in each timeslot.

V. MDMAR HEURISTICS

We have proven that the MDMAR problem is NP-Complete, hence no efficient optimal algorithm exists for it unless $P = NP$. In this section, we design heuristics for the two path schemes: the fixed path scheme and the various path scheme. Firstly, we propose a heuristic algorithm to minimize the total cost of the multicast tree based on the dynamic timeslot-based approach in the overview. Then, we give the greedy scheduling versions of the heuristic algorithm for the two path schemes. Finally, we use the simulated annealing (SA) to find the approximate solution.

A. OVERVIEW

One of the main problems of resource reservation for dynamic multicast is to optimize the cost of resource reserved in the multicast delivery tree, which has proven to be NP-complete. In this paper, we propose a heuristic algorithm to calculate the minimum cost of the multicast tree based on the dynamic timeslot approach when the individual receiver's QoS requirement is heterogeneous. In this paper, we mainly consider the bandwidth requirement as the QoS requirement and the cost of the tree is denoted as the reserved bandwidth for the multicast tree. In this paper, we dynamically build multicast trees for every source node of each conference.

As the request r_k arrives, the AR system rounds down the start time st_k to the nearest time point whose index in T is assumed as l and rounds up et_k to the nearest time point whose index is assumed as h respectively. As the *ith* timeslot is denoted as (T_{i-1}, T_i), the index of active timeslots of r_k is from $l + 1$ to h.

The AR system then finds the multicast tree that starts from the same source node as the request r_k, and adds the destination node to the multicast tree in the shortest path. Specifically, the AR system inclemently builds the multicast tree and reserves sufficient bandwidth on the tree as the request arrives. To identify the different multicast trees, we store all the source nodes of the conference c_m in the set S_m. The source node s_k is assumed to be located at n_k in S_m.

We calculate the cost of each link in each active timeslot for the request r_k. The cost of the *jth* link in the *ith* active timeslot for the request r_k is calculated by (16), where the $b_{n_k,i,j}^{res}$ represents the reserved bandwidth on the *jth* link in the *ith* timeslot for the multicast tree starts from the same source node s_k and the same conference c_k as r_k. (17) ensures that the jth link can participate in the path calculation only when the cost is less than the available bandwidth of the link. $b^a_{j,i}$ is denoted as the available bandwidth on the jth link in the ith timeslot.

For the fixed path scheme, the AR system calculates the total cost of the link for the whole session time for the request r_k. (18) calculates the cost of the jth link for the request r_k for the whole session time based on (17). Finally, the AR system calculates the shortest path based on the cost of each link. Meanwhile, the reserved bandwidth on the path is updated in (19) as well as the available bandwidth in (20).

$$w_{k,j,i} = \max(0, b_k - b^{res}_{n_k,i,j}),\quad i \in [l+1, h],\ j \in [1, |E|],\ \forall r_k \in R. \tag{16}$$

$$w_{k,j,i} = +\infty, \quad w_{k,j,i} \le b^a_{j,i},\quad j \in [1, |E|], i \in [l+1, h],\ \forall r_k \in R. \tag{17}$$

$$w_{k,j} = \sum_{i=l+1}^{h} w_{k,j,i}, \quad j \in [1, |E|],\ \forall r_k \in R. \tag{18}$$

$$b^{res}_{n_k,i,j} = b^{res}_{n_k,i,j} + w_{k,j,i},\quad j \in [1, |E|],\ i \in [l+1, h],\ \forall r_k \in R. \tag{19}$$

$$b^a_{j,i} = b^a_{j,i} - w_{k,j,i},\quad j \in [1, |E|],\ i \in [l+1, h],\ \forall r_k \in R. \tag{20}$$

According to (16)-(20), the AR system should define a two-dimensional matrix $b^a[|E|, |T|]$, which stores the available bandwidth on each link in each timeslot. For each conference, the AR system should define a three-dimensional matrix $b^{res}[|S_m|, |T|, |E|]$, which stores the reserved bandwidth on each link in each timeslot for the multicast tree that starts from each source node in S_m.

B. THE GREEDY SCHEDULING ALGORITHMS FOR THE DYNAMIC MULTICAST

In this section, we present the detailed algorithms for the fixed path scheme and variable path scheme. To achieve a better performance, the AR system should consider not only the resource reservation problem but also the scheduling strategies. In this section, we use the greedy strategy to schedule the AR requests. The main idea of greedy scheduling strategy is that, before handling the requests, the set of conferences and the set of corresponding connection requests are sorted by the number of connection requests of the conferences descending.

Algorithm 1 Greedy Scheduling for the Fixed Path (GSFP)

Data: $G(V, E, B)$, time domain d, granularity g, C, R

1 $[T, b^a] \leftarrow Initialize(d, g, G)$;
2 $[C, R] \leftarrow SortByNumber(C, R)$;
3 **for** $m = 1; m \le |C|; m{+}{+}$ **do**
4     $S_m = \varnothing$;
5     $b^{res} = \{\}$;
6     $[Ttmp, btmp^a] \leftarrow [T, b^a]$;
7     $flag = 1$;
8   **for** $\forall r_k \in R$ *and* $c_k == m$ **do**
9   round down and up the start time and end time of r_k, and sort insert them into T;
10   locate the start time in T at index l and the endtime at h;
11   update b^a and b^{res} with the insert of the request times;
12   insert the source node s_k in S_m and identify the location as n_k;
13   update the b^{res} when s_k is inserted;
14   **for** $j = 1; j \le |E|; j{+}{+}$ **do**
15   **for** $i = l+1; i \le h; i{+}{+}$ **do**
16   calculate $w_{k,j,i}$ for the jth link in (16);
17   update $w_{k,j,i}$ in (17);
18   **end**
19   calculate $w_{k,j}$ in in (18);
20   **end**
21   obtain *Gtmp* from G and assign the cost $w_{k,j}$ to each link e;
22   calculate the shortes path p in *Gtmp* from s_k to d_k and return the shortest distance d;
23   **if** $d == +\infty$ **then**
24             $flag = 0$;
25   break;
26   **else**
27   **for** $\forall jth$ *link on* p **do**
28   **for** $i = l+1; i \le h; {+}{+}i$ **do**
29   calculate $w_{k,j,i}$ in (16);
30   update $b^{res}_{n_k,i,j}$ in (19);
31   update $b^a_{j,i}$ in (20);
32   **end**
33   **end**
34   **end**
35   **end**
36   **if** *flag==0* **then**
37         $[T, b^a] \leftarrow [Ttmp, btmp^a]$;
38   **end**
39 **end**

The greedy scheduling for the fixed path (GSFP) is shown in Algorithm 1. The Initialize function initializes the time array T and available matrixb^a in Line 1. The conference requests and corresponding connection requests are sorted by the number of the connection requests descending, and then the connection requests are processed sequentially. The outermost for-loop that covers Lines 3-39 handles each conference request. First, it initializes the parameters for each conference in Lines 4-5; then, it saves global parameters in temporary variables. This is because the conference will be accepted only when all the connection requests of the conference are admitted. The middle for-loop that covers Lines 8-35 handles each video stream connection request of the conference. The innermost for-loop that covers Lines 14-20 calculates the cost on each link for the whole session time. For-loop that covers

Lines 27-33 update the global resource matrices if the connection request is accommodated with sufficient bandwidth; otherwise, they are reassigned the previous value as shown in Lines 36-38.

The greedy scheduling for the variable path (GSVP)is shown in Algorithm 2. Different from the fixed path scheme, GSVP performs access control in each timeslot for each connection request. Here, we only show the access control process for each connection request, as shown in Algorithm 2. The omitted part is the same as the corresponding part of algorithm 1. The outermost for-loop that covers Lines 2-25 handles each connection request. The middle for-loop that covers Lines 4-21 performs access control in each active timeslot. The inner for-loop that covers Lines 5-8 calculates the cost $w_{k,j,i}$, and then it performs access control. If the request cannot be accommodated with sufficient bandwidth, it breaks from the current timeslot, as shown in Line 13 and then breaks from the current connection request, as shown in Line 23.

$$D = \lfloor \frac{d}{g} \rfloor. \tag{21}$$

Algorithm 2 Greedy Scheduling for the Variable Path (GSVP)

```
1  ....;
2  for ∀r_k ∈ R and c_k == m do
3      handle the request time and update the resource
       matrix as Lines 9-13 in Algorithm 1;
4      for i = l; i ≤ h; i + + do
5          for j = 1; j ≤ |E|; j + + do
6              calculate w_{k,j,i} for the jth link in (16);
7              update w_{k,j,i} in (17);
8          end
9          obtain Gtmp from G and assign the cost w_{k,j,i} to
           each link e;
10         calculate the shortes path p in Gtmp from s_k to
           d_k and return the shortest distance d;
11         if d == +∞ then
12             flag = 0;
13             break;
14         else
15             for ∀jth link on p do
16                 calculate w_{k,j,i} in (16);
17                 update b^{res}_{n_k,i,j} in (19);
18                 update b^a_{j,i} in (20);
19             end
20         end
21     end
22     if flag == 0 then
23         break;
24     end
25 end
26 ...;
```

$$Q = \begin{cases} \sum_{k=1}^{|R|} 2k+1, & |R| <= S \\ \sum_{k=1}^{S} 2k+1+\sum_{k=S+1}^{|R|} D, & |R| > S \; S=\lfloor \frac{D-1}{2} \rfloor. \end{cases} \tag{22}$$

Complexity: In the dynamic timeslot approach, the number of timeslots increases dynamically. Assume that the length of time domain is d and the granularity is g, then the maximum number of timeslots can be calculated in (21). The initial number of timeslot is 1 as the time domain is not partitioned. As the requests arrive, the time domain will be partitioned into smaller timeslots until the number of timeslots reaches D. For each request, at most two timeslots will be added. Therefore, the number of timeslots can be calculated in (22).

The complexity of GSFP mainly focuses on the three parts: the calculation of link cost on Lines 14-20 and the corresponding complexity is $Q \times |E|$; the path calculation for each request on Lines 21-22 and the corresponding complexity is $|R| \times |V|^2$, and the resource matrix update on the path on lines 27-33 and the corresponding complexity is $Q \times |E|$. Therefore, the complexity of GSFP is $O(Q \times |E| + |R| \times V^2)$.

GSVP performs access control in each timeslot while GSFP only performs one access control for all the timeslots for each request. The complexity of GSVP is $O(Q \times (E + |V|^2))$.

Obviously, the complexity of GSVP is higher than that of GSFP. However, the acceptance ratio of GSVP may be higher than that of GSFP. This is because GSFP computes a fixed path for the whole session time, which imposes stricter requirement for the link bandwidth and may result in failure to accommodate the sufficient bandwidth.

C. SIMULATED ANNEALING (SA)

In addition to the simple heuristics for the two path schemes discussed, we also implemented a meta heuristic algorithm for each scheme based on simulated annealing. The above two heuristics sort the conferences by the number of the connection requests descending. However, the descending sort may not always result in a global optimal solution. In fact, due to the advance reservation attribute, the conferences can be handled in any order. We can explore a number of conference sequence and obtain a better solution. This can be thought of as a combinatorial optimization problem. In 1983, S. Kirkpatrick *et al.* succeeded in introducing annealing into combinatorial optimization [43]. It is based on the similarity between the annealing process of solid substances in physics and the general combinatorial optimization problem. To get a low energy state, the material is first heated up, then cooled down slowly. A low energy state is an equilibrium state for the particles to try to reach. When the temperature is very high, the particles of a solid will wander randomly and rearrange themselves. As the descending of the temperature, the particles will make moderate changes.

We adapt SA to solve our problem. To adapt SA to a new combinatorial optimization problem, three problems need to be solved: 1) an objective function; 2) a cooling schedule; 3) a perturbation function. In our SA, the objective function is to maximize the number of the admitted connection requests. A typical cooling schedule $T_{iter} = \alpha T_{iter-1}$ is used in our SA. We switch the locations of any two conference requests and corresponding connection requests to perturb the current solution.

We design the simulated annealing algorithms for the fixed path and variable path (SAFP, SAVP), as shown in Algorithm 3. Firstly, we use the approximate solution obtained by GSFP or GSVP as the initial solution, which is as shown in Line 4. The annealing process stops until the temperature is below a threshold, or a maximum number of iteration is reached, which is as shown in Lines 6-31. At each temperature, a number of perturbations are made to the current solution, as shown in Line 11. If the new solution obtains a better result, the new solution is accepted, as shown in Lines 14-17; otherwise, the worse solution is accepted based on the Metropolis criterion. The probability distribution used is the Boltzmann distribution and is given as $e^{-\Delta/T_0}$ where Δ is the difference value between the current energy and the new energy, as shown in Lines 18-22. From the probability distribution, we can observe that a higher temperature will result in a higher probability to accept the worse solution; however, as the descending of the temperature, the probability decreases and it turns to a focused local search. For each perturbation, the best solution found so far is stored in the array *Obj*, as shown in Lines 23-27.

Algorithm 3 Simulated Annealing for the Fixed Path or Variable path(SAFP/SAVP)

```
1  initialize T_0, T_end, α, L, iterMax;
2  iter = 1;
3  count = 1;
4  [CurNum, curC, curR] = GSFP/GSVP(G, d, g, C, R);
5  Obj(count) = CurNum;
6  while T_0 ≥ T_end && iter ≤ iterMax do
7      iter = iter + 1;
8      i = 0;
9      while i < L do
10         count = count + 1;
11         [newC, newR] = perturb(curC, curR);
12         newNum = GSFP/GSVP(G, d, g, newC, newR);
13         dif = curNum − newNum;
14         if dif < 0 then
15             curC = newC;
16             curR = newR;
17             curNum = newNum;
18         else if e^(−dif/T_0) > rand then
19             curC = newC;
20             curR = newR;
21             curNum = newNum;
22         end
23         if curNum > Obj(count − 1) then
24             Obj(count) = curNum;
25         else
26             Obj(count) = Obj(count − 1);
27         end
28         i = i + 1;
29     end
30     T_0 = αT_0;
31 end
```

Complexity: the complexity of SA is directly related to the complexity of GSFP and GSVP. The number of iterations also takes effect on the complexity of SA. The perturbation operation takes constant time, and so does the current solution and energy update.

VI. SIMULATION AND EVALUATION

In this section, we evaluate the performance of our heuristics for the MDMAR problem. Firstly, we compare them to the optimal results provided by the ILP models as a benchmark on a small network. Then we make an extensive evaluation for the heuristics on a campus network from the aspect of the traffic load, the available bandwidth, and the time granularity size.

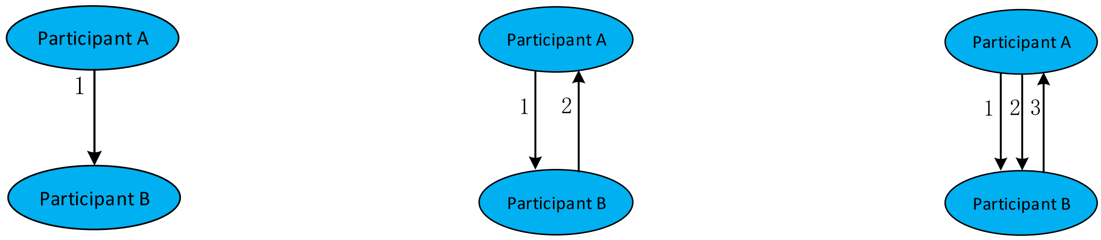
We use the acceptance ratio as a measure to assess the algorithm performance, which is expressed as in (23), where $|SR|$ indicates the number of successful requests and $|R|$ indicates the total number of connection requests.

$$Acceptance\ ratio = \frac{|SR|}{|R|}. \tag{23}$$

A. SIMULATION SETUP

RFC 4597 [6] describes a set of basic and advanced conference scenarios. Considering the predictability of the number of video streams of a conference, we use three reservable conference scenarios for the simulations. One scenario is called a "Lecture" conference. This conference scenario enables a lecturer to present a topic seen and heard by other participants, who cannot be seen and heard by other participants. Another scenario is called a "Discussion" conference, in which every participant can be seen or heard by all the other participants. The third scenario is called a "Presentation and Q & A" conference, which consists of two phases. When the phase of the presentation is finished, it starts a "Q & A" phase. The presentation phase is like the "Lecture" scenario. The "Q & A" phase is like the "Discussion" scenario.

Three scenarios produce the different number of video streams as shown in Fig. 5. Supposing that it is a two-party conference. (a) It produces one video stream from participant A to B for a "Lecture" conference. Participant A is the lecturer who sends the video stream and participant B can only receive the video stream. (b) It produces two video streams between participants A and B for a "Discussion" conference. A and B can send video stream to each other. (c) It produces three video streams between participant A and B for a"Presentation and Q & A" conference. Participant A first



(a) Lecture scenario (b) Discussion scenario (c) Presentation and Q & A scenario

FIGURE 5. Different conference scenarios and corresponding video stream connection requests.

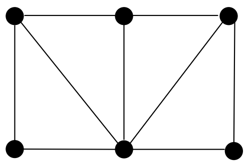


FIGURE 6. The smaller network for the evaluation of the ILP algorithms and the heuristics.

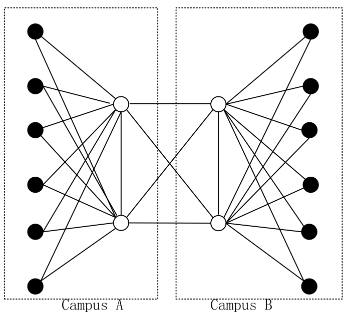


FIGURE 7. A typical campus topology.

sends a video stream to B in the "Presentation" phase. When the phase is finished, they send video streams to each other.

Because of the limited scalability of the ILP algorithm, a smaller network, as shown in Fig. 6, is used for the comparison of the heuristics to ILP while a larger typical university network is used to evaluate the performance of our heuristics, as shown in Fig. 7. The university has two campuses: Campus A and B, the solid circles can be connected to the terminal, and the hollow circles are the intermediate switches connected in a full mesh topology as shown in Fig. 7. For the ILP model, the conferences are assumed to be four-party, and four nodes are randomly selected from the smaller network. The conferences on the larger network topology are assumed to be six-party, and the endpoints are averagely distributed in the two campuses.

The video stream connection requests are generated in the conference unit. One of the three conference scenarios is randomly selected. And then, according to the conference scenario, it produces the corresponding number of video stream connection requests. All the video stream connections are generated according to the Poisson traffic model. The start time of each connection follows a Poisson distribution with the average arrival rate λ. The holding time of each

TABLE 1. 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

connection follows the negative exponential distribution with an average of $\frac{1}{\mu}$. Hence, the number of connections 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 [44]. In this paper, we set the average holding time to a mean of 50 minutes as the educational films in [44]. When endpoints join the conference, the bit rate of the endpoints is randomly set according to Table 1, and the video coding formats are assumed to be H.264. For the SA algorithms, we set $T_0 = 10^{10}$, $T_{end} = 10^{-10}$, $q = 0.9$, $iterMax = 10$, and $L = 10$. The larger $iterMax$ and L do not result in any better performance.

We use Lingo v11.0 to solve the ILP and simulate the heuristics for the smaller network topology in a MATLAB R2015b environment run on Windows 10 with a 2-GHz Intel Core i5 and 8-GB RAM. Simulations only for the heuristics 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. Error bars denote the standard error.

B. ILP RESULTS

We first evaluate the performance of the heuristics by comparing them to the optimal solution from the ILP on a small six-node network as shown in Fig. 6.

1) IMPACT OF TRAFFIC LOAD

Fig. 8 compares the acceptance ratio of algorithms for the different traffic loads, where the granularity is 1 minute and the bandwidth of each link is 15Mbps. As can be seen in Fig. 8, with the increasing traffic loads, the percentage of acceptance ratio decreases. This is because there is not sufficient bandwidth to accommodate the increasing requests. We observe that the simulated annealing algorithms achieve almost the same performance as the ILP algorithms. Specifically, ILPFP

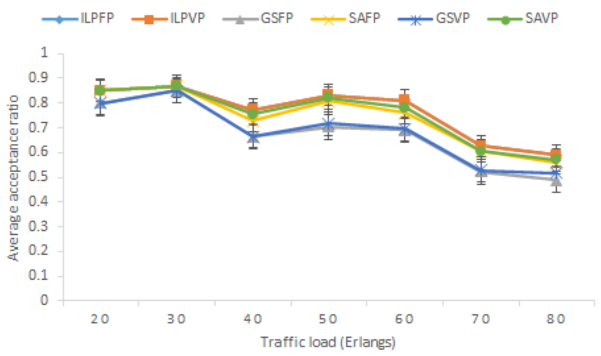


FIGURE 8. Impact of traffic load on acceptance ratio for ILP algorithms and the heuristics.

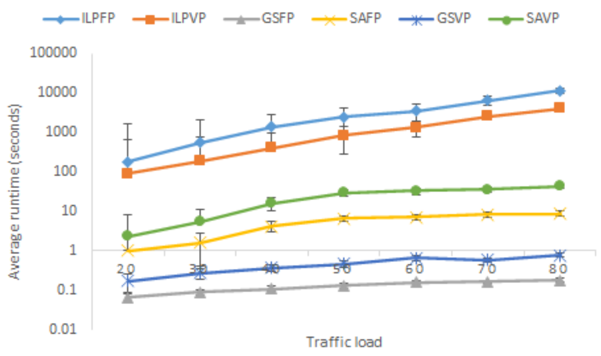


FIGURE 9. The runtime of the ILP algorithms and the heuristics.

and ILPVP outperform SAFP and SAVP within 3% and 2% respectively. When compared to the greedy algorithms, the simulated annealing algorithms show more effective. Specifically, SAFP and SAVP outperform GSFP and GSVP by up to 10% at 40 Erlangs and 80 Erlangs respectively.

Fig. 9 compares the corresponding runtime of the heuristics and ILP algorithms. As can be observed from this figure, the ILP algorithms are the most complex and runs slowest. Specifically, SAFP can be faster than ILPFP from 180 times up to 1000 times and SAVP can be faster than ILPVP from 25 times to 100 times. Although the greedy algorithms run much faster than the simulated annealing algorithms, they are much less effective than the simulated annealing algorithms in the acceptance ratio.

2) IMPACT OF AVAILABLE BANDWIDTH

Fig. 10 studies the impact of available bandwidth on the performance of the ILP algorithms and the heuristics, where the traffic load is 40 Erlangs and the granularity is 1 minute. With the increasing bandwidth varying from 3 to 21 Mbps, the acceptance ratio increases from 5% up to 95%. This is because there are more available bandwidth to accommodate the requests. From the figure, we can observe that the simulated annealing algorithms are able to achieve optimal or close to optimal solutions as the ILP algorithms. Specifically, ILPFP yields 6% better results at 12 Mbps, 5% better results at 9 Mbps, and 4.5 % better results at 15 and 18 Mbps than ILPFP while it yields close to optimal solutions as ILPFP

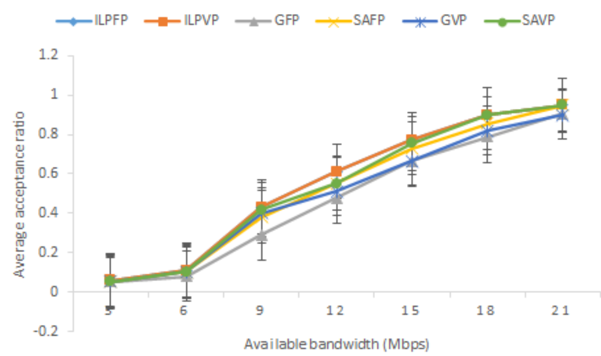


FIGURE 10. Impact of available bandwidth on acceptance ratio for ILP algorithms and the heuristics.

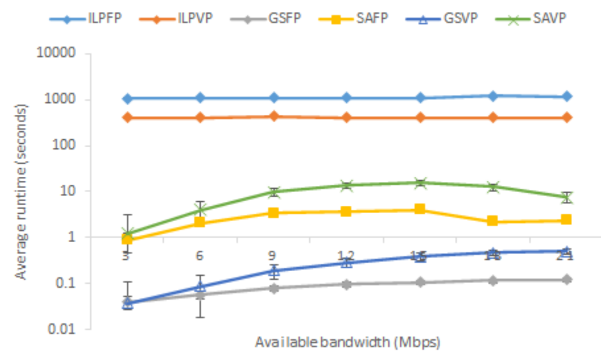


FIGURE 11. The run time of the ILP algorithms and the heuristics.

in other cases. By contrast, SAVP yields better results than SAFP. This is because the variable path is more flexible in paths than the fixed path. Specifically, ILPVP yields close to optimal solutions expect 6% better results at 12 Mbps. When compared to the greedy algorithms, the acceptance ratio of SAFP is up to 10 % higher than that of GSFP at 9 Mbps and SAVP can yields up to 9.5% better results at 15 Mbps than GSVP.

The corresponding runtime is depicted in Fig. 11. From the figure, we observe that, with the increasing available bandwidth, the runtime of the ILP algorithms increase slowly while that of the other algorithms increase steeply. This is because there are more bandwidth to accommodate the requests instead of rejecting them directly. Because the increase of bandwidth has less impact on the complexity of ILP algorithms, the increase of runtime of ILP is more slowly than that of other algorithms. As expected, the ILP algorithms consume the most runtime. Specifically, ILPFP takes 300 to 1200 times more runtime than SAFP and ILPVP takes 20 to 40 times more runtime than SAVP. When compared to the greedy algorithms, the simulated annealing algorithms consumes more time than them. Specifically, GSFP runs at most 40 times faster than SAFP at 9 Mbps and GSVP runs at most 50 times than SAVP at 9 Mbps. However, since the advance reservation is done offline and the simulated annealing algorithms are much more effective in acceptance ratio, it is reasonable for the simulated annealing algorithms to consume more time to improve the resource utilization.

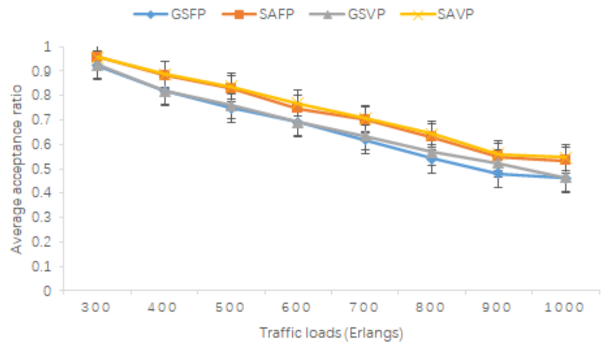


FIGURE 12. Impact of the traffic load on the acceptance ratio.

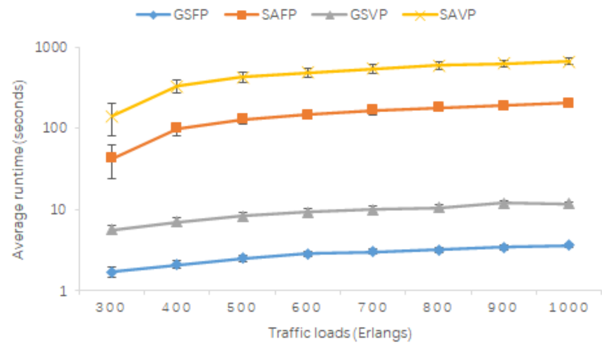


FIGURE 13. The average runtime of different traffic loads.

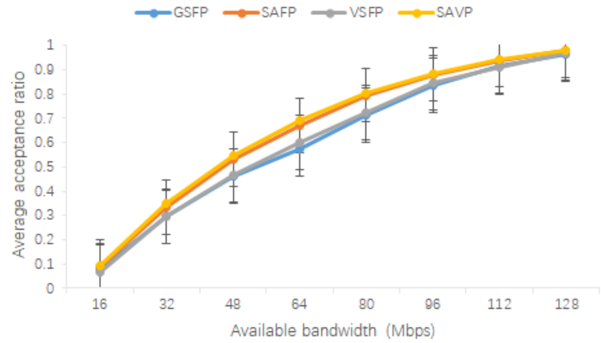


FIGURE 14. Impact of the available bandwidth on the acceptance ratio of the heuristics.

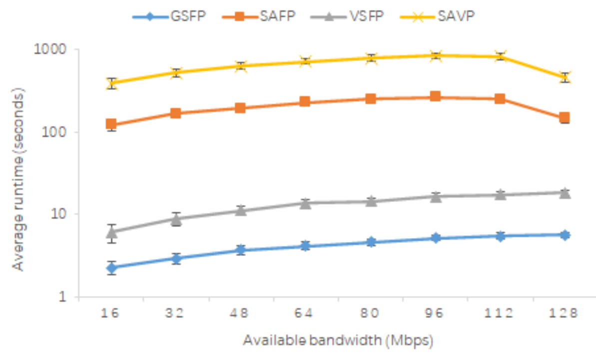


FIGURE 15. Impact of the available bandwidth on the runtime of the heuristics.

C. COMPARISON OF MDMAR HEURISTICS

We now assess the performance of the heuristics on the larger topology, as shown in Fig. 7, from the traffic load, available bandwidth and the size of granularity.

1) IMPACT OF TRAFFIC LOAD

Fig. 12 depicts the acceptance ratio regarding to the traffic load. We set the available bandwidth equal to 50 Mbps and the granularity equal to 20 minutes. As can be seen that in Fig. 12, with the increasing traffic load, the acceptance ratio decreases. This is because there are no sufficient available bandwidth for the increasing requests. The simulated annealing algorithms outperform the greedy algorithms for both fixed path and variable path. Specifically, the acceptance ratio of SAFP and SAVP is up to 10% and 8% higher than that of GSFP and GSVP respectively. GSVP outperforms GSFP by up to 5% higher.

The corresponding runtime is depicted in Fig. 13. As the traffic load increases, the runtime also increases. As expected, the simulated annealing algorithms take significant more runtime than the greedy algorithms. Since the advance reservation is done offline, it is worthwhile to sacrifice some runtime to increase the acceptance ratio. We also observe that the algorithms for the variable path take more runtime than that for the fixed path. Specifically, the algorithm for the variable path takes 3 times more runtime than the algorithm for the fixed path. The algorithms for the variable path is more effective in the acceptance ratio than that for the fixed path while they take more runtime. This is because the variable path scheme is more flexible in choosing paths than the fixed path scheme. However, due to performing the access control in each timeslot, the algorithms for the variable path takes more runtime than that for the fixed path.

2) IMPACT OF AVAILABLE BANDWIDTH

Fig. 14 studies the impact of different available bandwidth on the acceptance ratio of the heuristics. The available bandwidth varying from 16 to 128 Mbps, the granularity of 20 minutes and the traffic load of 1000 Erlangs are used. As the available bandwidth increases, the acceptance ratio increases too. This is because there are more available bandwidth to accommodate the requests instead of rejecting the requests directly. SAFP and SAVP achieve up to 10% and 9% higher results compared with GSFP and GSVP at 64 Mbps respectively. We also find that the algorithms for the variable path achieve almost the same acceptance ratio as that for the fixed path.

The corresponding runtime is depicted in Fig. 15. As can be seen in Fig. 15 that as the bandwidth increases, it takes more runtime to handle the same requests. As expected, The simulated annealing algorithms take more runtime than the greedy algorithms. Specifically, SAFP and SAVP take 50 times more runtime than GSFP and GSVP respectively. However, it may be worthwhile to increase the acceptance ratio by sacrificing some runtime, especially, in the case of running offline. From the figure, we observe that the algorithms for the variable path

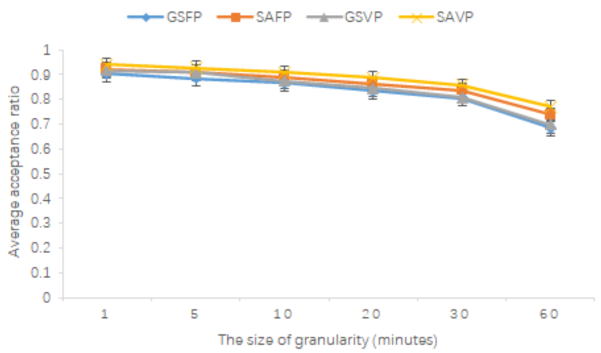


FIGURE 16. Impact of the granularity on the acceptance ratio of the heuristics.

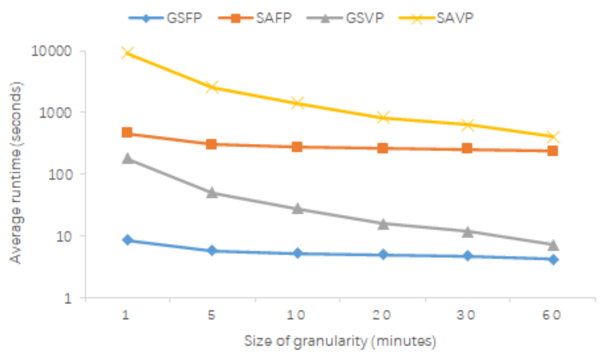


FIGURE 17. Impact of the granularity on the runtime of the heuristics.

take more runtime than that for the fixed path. Because the acceptance ratio of the algorithms for the variable is not better than that for the fixed path, it is not suggested to calculate the variable path in this case.

3) IMPACT OF GRANULARITY SIZE

Fig. 16 and 17 compare the influence of the granularity size varying from 1 minute to 60 minutes. In both figures, the traffic load of 1000 Erlangs and the available bandwidth of 100 Mbps are used. As shown in Fig. 16, a fine-grained granularity yields a higher acceptance ratio. However, although the fine-grained granularity optimizes the performance of the algorithms, the algorithms get more complexity and the runtime is significantly high as well, as shown in Fig. 17.

In Fig. 16, we find that SAVP yields 5% better results than GSFP at 1 minute. As the granularity size increases, SAVP can achieve 10 % better result than GSFP at 60 minutes. As expected, the simulated annealing algorithms outperform the greedy algorithms. Specifically, the acceptance ratio of SAFP and SAVP can be up to 6% higher and 8% than that of GSFP and GSVP at 60 minutes respectively. However, when comparing the complexity in Fig. 17, we observe that the simulated annealing algorithms get more complexity and run more slowly than the greedy algorithms. Due to the advance reservation is done offline, it is maybe worthwhile for the simulated annealing algorithms to take more runtime. The figure also shows that with the increase of time granularity, the runtime of the algorithms for the variable path decreases steeply while that for the fixed path keeps stable. With the increasing granularity size, the number of timeslots decreases. Because the runtime of the variable path is proportional to the number of timeslots while that of the fixed path keeps constant, the variable path shows a more steep trend than the fixed path, as shown in Fig. 17.

Based on the results, the granularity of 20 minutes optimizes the trade-off between effectiveness and efficiency. However, we cannot figure out the granularity of 20 minutes always yields the most optimal value for all the possible inputs.

VII. CONCLUSION

In this paper, we consider the situation where the participants may not arrive at the same time and stay until the end of the conference. To improve the resource utilization, every participant is required to provide not only the bandwidth but also the start time and holding time for the video stream connection. We investigated the problem of maximizing the number of the admitted dynamic multicast requests in the static advance reservation environment and showed that the problem is NP-complete. Furthermore, we considered the fixed path and variable path schemes for the requests, as well as the heterogeneous bandwidth reservation model. We formulated an ILP model for the small topology. Then, we designed four heuristics, which takes the heterogeneous bandwidth reservation model into account, for the large practical network topology. The performance in terms of acceptance ratio and runtime have been evaluated from the aspect of the traffic load, the available bandwidth and the granularity.

Our evaluation showed that the heuristics offer close-to-optimal solutions but much lower operational overhead when compared to the ILP algorithm. Specifically, the acceptance ratios of the simulated annealing algorithms are within 6% lower than the ILP algorithms respectively. When compared to the greedy algorithms, SAFP and SAVP outperform GSFP and GSVP up to 10%. Due to the scalability of ILP, the runtime of ILPFP and ILPVP can up to 1000 times and 100 times more than that of SAFP and SAVP respectively, and significantly more when compared to GSFP and GSVP respectively. When comparing the simulated algorithms and the greedy algorithms for the large network, the simulated annealing algorithms achieve improvement in terms of acceptance ratio while taking more runtime than the greedy algorithms. Specifically, SAFP and SAVP achieve up to a 10% improvement over GSFP and GSVP in terms of acceptance ratio respectively. The runtime for the simulated annealing algorithms is significantly increasing. However, the algorithms are computed offline, so the runtime of SA is reasonable given this case.

Future work is to study the impact of variable bandwidth, a different timeslot approach and the static and dynamic traffic models coexistence on the quality of the algorithm to improve the resource utilization.

REFERENCES

[1] Analytics, Global Workplace and Flexjobs. (2017). *2017 State of Telecommuting in the US Employee Workforce*. [Online]. Available: http://globalworkplaceanalytics.com/2017-state-of-telecommuting-in-the-us

[2] B. W. Reynolds. (2018). *Flexjobs 2018 Annual Survey: Workers Believe a Flexible or Remote Job Can Help Save Money, Reduce Stress*. [Online]. Available: https://www.flexjobs.com/blog/post/flexjobs-2018-annual-survey-workers-believe-flexible-remote-job-can-help-save-money-reduce-stress-more/

[3] 36Kr. (2020). When 200 Million People Start Video Conferencing. Video Conferencing. [Online]. Available: https://36kr.com/p/5290266/

[4] C. S. Lewis and S. Pickavance, *Selecting Mpls Vpn Services*. Indianapolis, IN, USA: Cisco Press, 2006.

[5] V. Cisco, "Cisco visual networking index: Forecast and trends, 2017–2022," Cisco, San Jose, CA, USA, White Paper 1, 2018.

[6] R. Even and N. Ismail, *Conferencing Scenarios*, document 4597, Aug. 2006.

[7] 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, 2012.

[8] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: A new resource reservation protocol," *IEEE Netw.*, vol. 7, no. 5, pp. 8–18, Dec. 1993.

[9] D. Ferrari, A. Gupta, and G. Ventre, "Distributed advance reservation of real-time connections," in *Proc. Int. Workshop Netw. Oper. Syst. Support Digit. Audio Video*. New York, NY, USA: Springer, 1995, pp. 16–27.

[10] J. Zheng and H. T. Mouftah, "Routing and wavelength assignment for advance reservation in wavelength-routed WDM optical networks," in *Proc. IEEE Int. Conf. Commun. Conf. (ICC)*, 2002, pp. 2722–2726.

[11] 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.

[12] K. T. Phan, J. Moulierac, C. N. Tran, and N. Thoai, "Xcast6 treemap islands: Revisiting multicast model," in *Proc. ACM Conf. CoNEXT Student Workshop*, 2012, pp. 33–34.

[13] C. Pornavalai, D. Chakraborty, G. Chakraborty, and N. Shiratori, "Dynamic multicast routing in advance resource reservation environment," in *Proc. 7th Int. Conf. Parallel Distrib. Syst.*, 2000, pp. 547–552.

[14] B. M. Waxman, "Routing of multipoint connections," *IEEE J. Sel. Areas Commun.*, vol. 6, no. 9, pp. 1617–1622, 1988.

[15] P. Pavarangkoon, A. Gunabhibal, C. Pomavalai, and R. Varakulsiripunth, "An efficient dynamic multicast routing algorithm with advance resource reservation awareness," in *Proc. 6th Int. Conf. Adv. Commun. Technol.*, 2004, pp. 651–655.

[16] B. B. Bista and G. Chakraborty, "Resource reservation with session time in multicast routing," in *Proc. 16th Int. Workshop Database Expert Syst. Appl. (DEXA)*, 2005, pp. 111–115.

[17] 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.

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

[19] 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.

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

[21] 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.

[22] E.-Z. Yang, L.-K. Zhang, Z. Yao, and J. Yang, "A video conferencing system based on SDN-enabled SVC multicast," *Frontiers Inf. Technol. Electron. Eng.*, vol. 17, no. 7, pp. 672–681, Jul. 2016.

[23] S.-H. Shen, "Efficient SVC multicast streaming for video conferencing with SDN control," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 2, pp. 403–416, Jun. 2019.

[24] M. Barshan, H. Moens, J. Famaey, and F. De Turck, "Deadline-aware advance reservation scheduling algorithms for media production networks," *Comput. Commun.*, vol. 77, pp. 26–40, Mar. 2016.

[25] C. Zhang, X. Huang, G. Ma, and X. Han, "A dynamic scheduling algorithm for bandwidth reservation requests in software-defined networks," in *Proc. 10th Int. Conf. Inf., Commun. Signal Process. (ICICS)*, Dec. 2015, pp. 1–5.

[26] 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.

[27] 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.

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

[29] V. Firoiu and D. Towsley, "Call admission and resource reservation for multicast sessions," in *Proc. 96th Conf. Comput. Commun.*, 1996, pp. 94–101.

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

[31] H. Soni, W. Dabbous, T. Turletti, and H. Asaeda, "Scalable guaranteed-bandwidth multicast service in software defined ISP networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–7.

[32] J.-J. Kuo, S.-H. Chiang, S.-H. Shen, D.-N. Yang, and W.-T. Chen, "Dynamic multicast traffic engineering with efficient rerouting for software-defined networks," in *Proc. IEEE INFOCOM - IEEE Conf. Comput. Commun.*, Apr. 2019, pp. 793–801.

[33] T. D. Wallace, A. Shami, and C. Assi, "Scheduling advance reservation requests for wavelength division multiplexed networks with static traffic demands," *IET Commun.*, vol. 2, no. 8, pp. 1023–1033, 2008.

[34] M. Moharrami, A. Fallahpour, H. Beyranvand, and J. A. Salehi, "Resource allocation and multicast routing in elastic optical networks," *IEEE Trans. Commun.*, vol. 65, no. 5, pp. 2101–2113, May 2017.

[35] W. Khallef, S. Durand, and M. Molnár, "ILP formulation of the exact solution of multi-constrained minimum cost multicast," *Comput. Netw.*, vol. 135, pp. 160–170, Apr. 2018.

[36] C. P. Low and N. Wang, "An efficient algorithm for group multicast routing with bandwidth reservation," *Comput. Commun.*, vol. 23, no. 18, pp. 1740–1746, Dec. 2000.

[37] C. P. Low, N. Wang, and J. M. Ng, "Dynamic group multicast routing with bandwidth reservations," *Int. J. Commun. Syst.*, vol. 15, no. 8, pp. 655–682, Oct. 2002.

[38] Z. Liao, L. Zhang, and J. Deng, "Research on framework for enterprise video conference system based on SDN," *J. Huazhong Univ. Sci. Tech.*, vol. 44, no. 1, pp. 204–209, 2016.

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

[40] Z. Liao and L. Zhang, "Elastic timeslot-based advance reservation algorithm for enterprise video conferencing systems," *IEEE Access*, vol. 8, pp. 5104–5120, 2020.

[41] A. Lodi, S. Martello, and M. Monaci, "Two-dimensional packing problems: A survey," *Eur. J. Oper. Res.*, vol. 141, no. 2, pp. 241–252, Sep. 2002.

[42] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA, USA: Freeman, 1979.

[43] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[44] 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, China. Her research interests include computer network communication, video conference service quality assurance, and software defined networking.

LING ZHANG (Associate Member, IEEE) 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 Profess or with the School of Computer Science and Engineering, SCUT. His research interests include computer network communication, network security, high-performance computing, and signal processing.