Frontiers of Information Technology & Electronic Engineering
www.zju.edu.cn/jzus; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

A video conferencing system based on SDN-enabled SVC multicast*

En-zhong YANG, Lin-kai ZHANG, Zhen YAO, Jian YANG‡
(*School of Information Science and Technology, University of Science and Technology of China, Hefei 230001, China*)
E-mail: ezhyang@mail.ustc.edu.cn; pkpk8@mail.ustc.edu.cn; yaozhen1@mail.ustc.edu.cn; jianyang@ustc.edu.cn

Received Mar. 14, 2016; Revision accepted June 19, 2016; Crosschecked June 25, 2016

Abstract: Current typical video conferencing connection is bridged by a multipoint control unit (MCU), which may cause large delay and communication bottleneck for the whole system. With the development of network technology, a video conferencing system can be implemented based on software-defined networking (SDN), which makes the service controllable and improves the scalability and flexibility. Additionally, a video encoding method called scalable video coding (SVC) can also help. In this paper, we propose a video conferencing architecture based on SDN-enabled SVC multicasting, which discards the traditional Internet group management protocol (IGMP) and MCU. The system implements SVC multicast streaming to satisfy different device capabilities of various conference terminals. The SDN controller is responsible for dynamically managing and controlling the layers of a video stream when a conference member faces network congestion. Also, a conference manager is designed to facilitate the management of the conference members. Experimental results show that our system can not only provide a flexible and controllable video delivery, but also reduce the network usage while guaranteeing the quality of service (QoS) of video conferencing.

Key words: Software-defined networking (SDN), Multicast, Scalable video coding, Video conferencing system
http://dx.doi.org/10.1631/FITEE.1601087 **CLC number:** TP393; TN919.8

1 Introduction

With the rapid development of society, people are no longer satisfied with simple voice and text communications. They are more inclined to multimedia communications with voice, text, images, and videos. As one of the most advanced communication technologies, video conferencing allows two or more locations to communicate by simultaneous two-way video and audio transmissions (Qiu *et al.*, 2002). Video conferencing has the unique advantages of increasing communication efficiency, reducing travel costs, and improving management effectiveness. However, with continuous increase in the number of users, how to provide high delivery quality and save network bandwidth becomes a challenge.

Video conferencing systems are deployed mainly by means of a multipoint control unit (MCU) (Willebeek-LeMair *et al.*, 1994). An MCU is a bridge that interconnects calls from several sources. All parties call an MCU, or the MCU can also call the parties that are going to participate in the conference (Fig. 1). It is simple and convenient to maintain the user status. However, because all terminals send control messages and videos to the MCU, and MCU sends videos back to all terminals, the MCU solution needs high network bandwidth and may cause large delays. The MCU may become the bottleneck with a heavy handling burden. It is hard to ensure

---

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 61573329 and 61233003), the Youth Innovation Promotion Association CAS, and the Fundamental Research Funds for the Central Universities, China

ORCID: En-zhong YANG, http://orcid.org/0000-0001-7592-7319

high-quality delivery, since its single function limits the scalability and reduces reliability.

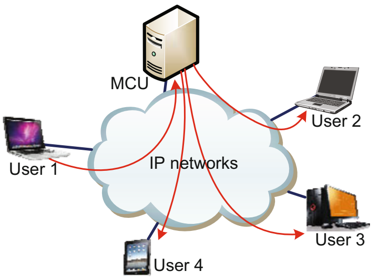

Fig. 1 MCU-based solution

Unicast is a way of sending content to a single receiver, but it does not consider that others may also need the same content. By contrast, multicast intends to deliver the same content to a group of receivers, thus avoiding the waste of network resource (Li *et al.*, 1999). Considering the features of a video conferencing system, multicasting is an efficient solution for conveying the video content. Some MCUs can support the Internet protocol (IP) multicast transmission mode. However, an IP multicast-based video transmission over the Internet is not widely applied. The reason for this is that the IP multicast requires transporting routers on the whole path supporting multicast, which induces additional configuration and security considerations, thus making the commercial Internet service providers (ISPs) reluctant to deploy native IP multicast. For commercial multimedia services, access control is necessary for providing a basic service-level security to ensure that data streams are sent only for legitimate receivers and requesting routers. In IP multicast, however, anyone joining the multicast group can obtain the multicast stream. This implies that IP multicast does not support an access control mechanism, and cannot be controlled and managed over the network.

To ensure high-quality video delivery and save the network bandwidth, a video encoding method called scalable video coding (SVC) (Schwarz *et al.*, 2007) can help. SVC is an extension of the H.264/AVC standard and keeps the advantages of H.264, i.e., high compression ratio and better image quality. SVC further provides three types of scalabilities: spatial, temporal, and quality scalabilities. By applying a layered approach, the encoded SVC bit stream consists of a base layer providing the basic quality and one or more enhancement layers. For example, the base layer provides the content quality needed for mobile devices; additional enhancement layers provide high-definition quality. The variety of conference member devices with different capabilities ranges from cell phones with small screens and restricted processing power to high-end PCs with high-definition displays. SVC provides bit-stream scalability for graceful degradation transmission or adaptation, thus fulfilling the customized requirements like adapting to the heterogeneous device capabilities and heterogeneous networks. However, SVC video streaming is still not widely applied in practical video transmission, although it has bit-rate adaptation ability. The basic reason behind this is that routers in the current network are transparent for SVC streaming applications, which disables in-network bit-rate adaptation for a graceful degradation of the transmission.

In recent years, software-defined networking (SDN) (Sezer *et al.*, 2013) has been fast emerging as a promising network technology for building next-generation services and networks. OpenFlow (McKeown *et al.*, 2008) is a standard communication protocol that enables SDN. It decouples the control plane and data plane of a network node, thus bringing out several nice features. The SDN controller can obtain the whole network's information and manage the network easily. It is convenient to deploy multicast over SDN without any distributed multicast routing protocols. Moreover, since the SDN controller can observe the link status, when conference members suffer from annoying congestion, the controller can configure the switches to forward fewer enhancement layers to ensure the fluency of video. When the network condition gets better, the number of enhancement layers will be increased to obtain a better quality. That is, SDN makes the deployment of SVC in-network adaptation transmission possible.

In this paper, we propose an SDN-enabled SVC multicast solution for multi-party video conferencing systems. Our approach discards the traditional Internet group management protocol (IGMP) (Fenner, 1997) and MCU to obtain a better performance, and deploys multicast by just using unicast addresses and uses a centralized management system to maintain the relationships between hosts and groups. Furthermore, to implement multicast over SDN, the service provider needs only to program in the SDN controller, and then the controller sends flow entries to the OpenFlow switch. The OpenFlow switch forwards the network packets following the rules of the flow table in the way of multicast. This avoids the complex network configuration in the current traditional network. SDN-enabled multicast can also provide an access control mechanism and enable the implementation of multicast authentication, authorization, and accounting (AAA) integration for user authentication in a multicast context. The SDN controller can create multicast distribution trees according to the locations of the participants. A conference member is not only a multicast tree leaf of one multicast group but also a multicast tree root of another multicast group. A conference member sends SVC-encoded video streaming to the network so that other members can receive different numbers of SVC layers to obtain different video quality. Using the controllable layered multicast video approach over SDN, the proposed system can save the network bandwidth and reduce delay effectively, as well as guarantee the quality of video conferencing for different users.

2 Related work

Research on the design of video conferencing systems has been going on for many years. Recently, Ng *et al.* (2014) proposed a P2P-MCU approach to support a multi-party WebRTC conference, which can reduce CPU usage and bandwidth consumption. Feng and Wu (2012) designed a cloud-based video conference multi-gateway system to meet the needs of a high-definition video conference system. This system can control a variety of virtual gateway resources based on physical devices; thus, multimedia gateways could be deployed dynamically. Zhang *et al.* (2014) proposed a ping-based clustering algorithm to determine the network topology and fine-tune the server locations with the objective of reducing end-to-end delays. Integrating SVC into MCU is state-of-the-art nowadays, and has caught the attention of enterprises and research institutes. Given the high scalability and flexibility of the SVC introduced above, it may be an effective solution to relieve MCUs from the stress of the bandwidth with SVC. In the case where there are a few conference members applying for accessing the MCU, since the bandwidth resource is enough for each participant, additional enhancement layers can be delivered by the MCU with little loss to provide high image quality. Otherwise, if there are too many members participating, the MCU will reduce the number of enhancement layers to guarantee the fluency and reliability of the communication. Some vendors like Radvision (Avaya, 2016) and Polycom (Polycom, 2014) have proposed several solutions for applying SVC in MCUs to make their products competitive. These systems mentioned above are based on the traditional networks without considering the feature of SDN. SDN is an active research topic in the area of networking, and is very attractive for the academia and industry. Zhao *et al.* (2014) used SDN-enabled multicasting to facilitate multi-party video conferencing application, which can provide better video delivery compared with the conventional MCU-based solution. However, they used single-layer video multicasting to deploy the system, which lacks the support of in-network bit-rate adaptation. To use a layered multicast approach to implement the system, a mechanism of supporting heterogeneous terminal devices and SVC video in-network transmission adaptation is needed.

Civanlar *et al.* (2010) described an SVC video streaming architecture for supporting quality-of-service (QoS) flows in the OpenFlow environment. Egilmez *et al.* (2013) extended it to optimize forwarding decisions at the control layer to make dynamic QoS supporting possible. However, these works focused on unicast-based SVC transmission. SVC transmission with multicast can be classified as a layered multicast issue. The receiver-driven layered multicast (RLM) was introduced by McCanne *et al.* (1996) to transmit layered signals over heterogeneous networks using receiver-driven adaptation. Zhang and Li (2002) proposed a receiver-driven, router-assisted layered multicast protocol (RALM) to achieve enhanced performance. Nevertheless, these works are based on IP multicast over traditional networks to design the video streaming scheme, lacking service and network control.

In this paper, we use a controllable layered multicast approach to implement video conferencing over SDN. The proposed system can save the network bandwidth and reduce the delay effectively. Moreover, it uses the SVC in-network video layer adaptation method to make dynamic QoS supporting possible. To our knowledge, this is the first work of designing a video conference system using SDN-enabled layered video multicast.

3 System design

The design goals for the architecture are as follows: obtain a stable and clear system while saving the bandwidth, reducing delay, and guarantee the quality of the video conference for different users. In designing the video conferencing system, the major challenges are how to manage the meeting, how to identify the SVC video stream, how to build multicast trees, and how to guarantee QoS. The overall architecture of the proposed video conferencing system is as shown in Fig. 2. It uses a controllable layered multicast mode to minimize network bandwidth usage. The conference members can subscribe to different video layers according to the device capability. They will at least receive a base layer to ensure the smooth video playback and one or more enhancement layers to increase the video quality. For instance, conference members with small-screen smartphones just need to receive the base layer stream, while others with high-definition displays can obtain more enhancement layer streams. As shown in Fig. 2, the PC and smartphone are connected with the same OpenFlow switch, but the PC may receive four layers of video sent from the notebook, while the switch may forward just one layer to the smartphone. Each SVC layer is defined as a flow, and the video sender and receivers form a group. For a video conference, the number of groups equals the number of video members who send video streams to other participants. The group information is maintained in the conference manager, which also maintains the information of forwarding layers according to the heterogeneous devices with different capabilities. The SDN controller can configure switches to collect the information of network nodes, so that the system can adaptively decrease the number of video layers to avoid congestion. If the network link state becomes good enough, the member terminal will be able to receive more enhancement layers to obtain a higher video quality.

There are four key components of the system, as shown in Fig. 3: the terminals, the conference manager, the controller, and OpenFlow switches. OpenFlow switches and the links between them constitute the infrastructure layer of the system. The control layer is based on an OpenFlow controller, which contains network information on the infrastructure layer. The southbound interface is the OpenFlow protocol, which is used for communication with OpenFlow controllers. The conference manager and parts of modules in the controller implement the application layer functionality. The modules in the OpenFlow controller are used by the conference manager from the application layer through the northbound interface to implement cross-layer functionality. Details about the components of the proposed system are described below.

3.1 Terminals

There are two types of terminals in the conference system: the president terminal and the member terminal. The president terminal, as the organizer and primary spokesman of the conference, creates a conference and manages the related conference

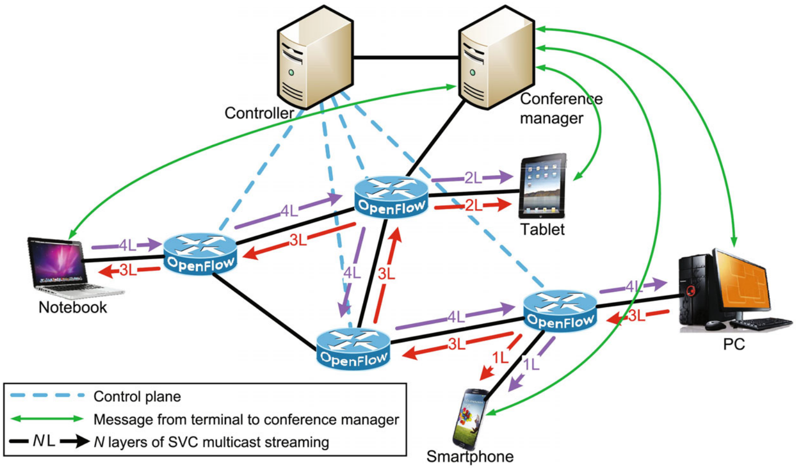

Fig. 2 SDN-enabled SVC multicast solution of the video conferencing system

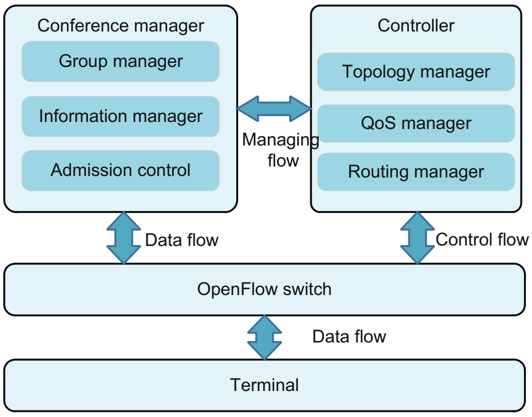

Fig. 3 Design of the SDN-enabled SVC multicast video conferencing system

configuration. A member terminal is an ordinary conference member who participates in the conference. If a member terminal applies to speak, it converts to the president terminal with the corresponding competence after being allowed. Each conference terminal includes mainly a multimedia processing module, which supports SVC encoding and decoding. The conference member terminals can be smartphones, tablets, notebooks, HDTVs, and so on. According to the device capability, different terminals can subscribe to different suitable video layers.

Video and audio are converted into digital format through the coding device. The encoded SVC bit stream consists of a base layer to provide the basic video quality and several enhancement layers to provide better quality. When a terminal sends the encoded video, different layers of the video are transmitted at different user datagram protocol (UDP) source ports. The conference manager records these source ports and notifies them to the controller, so that the OpenFlow switches can be content-aware. In other words, the OpenFlow switches can identify to which terminal and which video layer the data packets belong by matching the IP and UDP/TCP (Transmission Control Protocol) source port field.

When new users request the conference manager to join a conference, the member terminals will report the device capability information to determine the number of transferring layers so that they can receive suitable video layers forwarded by the OpenFlow switches from the layered SVC video source. The receiver device decodes the receiving digital stream into a form that can be displayed and heard.

3.2 Conference manager

The conference manager is established to facilitate the management of conferences. It provides interactive communication service with terminals and the controller. When a member terminal applies to the conference manager for joining a conference, its related information will be maintained. The conference manager will process and maintain the data as an information center. Then relevant information will be sent to the controller to do some network-level work, such as constructing multicast trees. Specifically, we construct a multicast tree for each video layer to realize the framework for SVC multicast streaming. The conference manager manages the corresponding events through three basic modules: group manager, information manager, and admission control.

1. Group manager module: IP multicast uses IGMP to manage group members, which needs a complicated network configuration. The SDN-enabled SVC multicast system abandons the traditional IGMP, and group members are maintained in the conference manager instead. The group manager module maintains a correspondence list of the conferencing group and its members. Each video sender and the receivers form a group with a unique group ID (GID). The list preserves the receivers' MAC and IP addresses, defined as

$$\mathrm{Sender[GID]} = [(\mathrm{mac}_1, \mathrm{ip}_1), (\mathrm{mac}_2, \mathrm{ip}_2), \ldots],$$

where mac_i and ip_i $(i = 1, 2, \ldots)$ denote the MAC and IP addresses of receiver i, respectively. If a member terminal has sufficient permission to join the conference and wants to watch the video of a participant, when the corresponding GID exists, its related information will be added to this list.

2. Information manager module: This module is responsible for managing the configuration of each conference and member information. From the aspect of a conference, the stored information includes the list of conferences, group members of a conference, group IDs, conference properties, and so on. From the aspect of members, the stored information includes capability of the user terminals for subscribing to suitable video layers, the member ID to identify receivers for each video layer, user permission to join a conference, the IP/MAC address to support network communications, etc.

3. Admission control module: The admission control module is responsible for the management process of a new member joining or leaving a conference and maintaining the related information. For instance, when a member applies to join or leave a conference, its access information (access the first time, access once more, or leave) will be added to the conference manager for maintenance, so that the controller can analyze and process the information with corresponding events. Here, we define a dictionary host[mac] = GID to maintain the relationship between a member terminal and a conference so that we can judge whether the member is accessing for the first time or accessing again. Then we use the heartbeat mechanism to determine whether the user is still online. Fig. 4 illustrates the processes of the admission control module when a new member requests to join the conference. At first, only hosts A and C are in the conference, and then host B applies for joining the conference group.

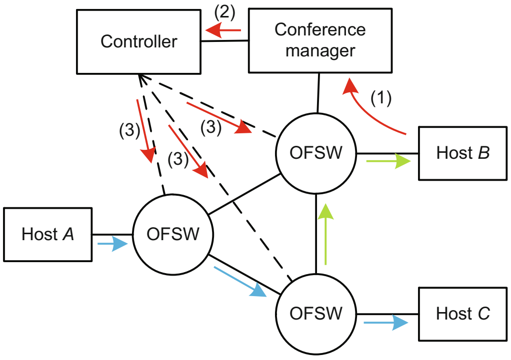

Fig. 4 Admission control management process (OFSW: OpenFlow switch)

As shown in Fig. 4, the process can be described as follows: (1) Host B sends a message to the conference manager applying to join the conference group. (2) The conference manager checks the permission first. If it is allowed, the conference manager will update the list of the conference group members and send the corresponding message to the controller. (3) Through the corresponding events management system, the controller updates the group members, calculates a new multicast tree, and sends flow table entries to the switches.

3.3 Controller

In the SDN-enabled SVC multicast system, the SDN controller will make high-level routing decisions. Unlike traditional IP multicast, the centralized controller can obtain the whole network's topology status from OpenFlow switches and multicast group member information from the conference manager, so it is convenient and efficient to set up multicast trees. Here, the controller can build multiple multicast trees for the same video of different layers. When suffering from network congestion, the controller is capable of configuring the OpenFlow switches to dynamically decrease the number of SVC layers through the corresponding multicast tree to avoid annoying playback interruptions. Current controllers in the mainstream include NOX, POX, Floodlight, etc. Among them, we choose the POX controller (Open Networking Lab, 2015) to set up our system. It is composed of the following three parts:

1. Topology manager module: This module maintains the network topology information. Link discovery is the key to obtaining the entire network information. When a switch is added to the SDN environment, the controller can obtain link information between the switch and other connected devices by a protocol called the link layer discovery protocol (LLDP). At first, the controller will send a 'packet-out' message to send LLDP packets to all the OpenFlow switches. The message commands the OpenFlow switch to send LLDP packets to the connected devices through all its ports. As the switch has no specific flow table entries for processing an LLDP message, all packets will be sent to the controller through a 'packet-in' message. After receiving the 'packet-in' message, the controller will analyze the packets and create a link between two switches in the table. Therefore, the controller will be able to create a complete view of the network topology.

2. QoS manager module: The QoS manager basically facilitates the system to achieve in-network adaptation of scalable video transmission to address the fluctuating link quality. This module provides a link and port status monitoring function via collecting 'statistics' messages supported by the OpenFlow protocol, and uses 'statistics' messages to detect the congestion event. As soon as a change in link quality is detected, the routing manager module is triggered to determine an appropriate number of video layers to be transmitted over this link. This system implements a threshold-based layer selection strategy. For instance, if a link and port's usage ratio increases to

a value higher than the threshold, the QoS manager triggers the routing manager module to decrease the number of layers transmitted over the overused link. Otherwise, when detecting plentiful bandwidth, it triggers the routing manager module to increase the number of transmitted layers, and thus the terminal obtains a higher quality of experience (QoE).

3. Routing manager module: The routing manager module constructs multicast trees and sends flow table entries to OpenFlow switches to deploy the logical routing path. When a conference begins, the topology and terminal list in the group manager module will be updated. As our system employs a layered multicast architecture to deliver SVC video stream, we construct a multicast tree for each video layer. Each video layer can identify receivers by a member ID maintained in the conference controller. Relying on the topology maintained in the controller, the conference manager locates the source and receivers, and applys a specific multicast tree creation algorithm, such as the PRIM algorithm, to construct a minimum spanning tree (MST) for each SVC layer. Then the system will map these logic multicast trees into OpenFlow rules. Finally, the SDN controller configures these forwarding rules into the flow table of switches. When the system suffers from network congestion, this module can dynamically adjust the number of video layers and transmit data packets through the corresponding multicast tree. Then the receiving terminal decodes the receiving data stream that can be seen and heard on the monitors.

3.4 OpenFlow switches

An OpenFlow switch is responsible for the data packets forwarding in the data plane and can rewrite flows' addresses, under the guidance of the controller. A transmission path for each conference video layer's flow can be dynamically adjusted by the controller configuring the OpenFlow switches via the OpenFlow protocol.

3.5 Discussion

It is simple to deploy such an SDN-enabled conferencing system from a technical perspective. However, in common practice, the SDN controller is owned by the ISPs, while video conferencing service providers need to program on the network controller. Thanks to SDN supporting network virtualization, it is possible for ISPs to give each application service provider its own network topology and control over its traffic flow (Drutskoy *et al.*, 2013). By means of network virtualization, video conferencing service providers may implement the proposed system in the controller of the network slice, which makes the network scalable and flexible.

4 Experimental results

4.1 Experimental environment setup

In this work, we refurbished Netgear WNDR3800 routers with the OpenWrt (OpenWrt Developer Team, 2015) system and made the routers OpenFlow switches supporting OpenFlow version 1.0. In our experiment, we used 16 such OpenFlow switches to build an OpenFlow network to test our system's performance. The network topology is as shown in Fig. 5. Each switch's wide-area network (WAN) port was configured as a console port, which was linked to the SDN controller. The four LAN ports were linked to other OpenFlow switches or member terminals to forward video streaming data. The WLAN interface was also configured to support the OpenFlow protocol so that wireless devices, such as smartphones or notebooks, can also access the network via Wi-Fi.

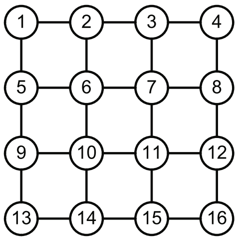

Fig. 5 Topology of the experiment network

We chose POX as our controller, running on a Linux server with two Intel Xeon E5-2620 CPUs, 64 GB RAM, and two network interface cards (NICs). The first NIC was connected to a standard hub, which was connected with all the OpenFlow switches' console port to constitute an out-of-band control network. The second was connected to the conference manager to transmit the information needed. Before the experiment, we programmed the POX for implementing each module in the controller.

The conference manager operated on another Linux server with two NICs, which were respectively connected to the controller and OpenFlow switches to collect information about conferencing sessions, member terminals, and multicast groups. These data structures would be maintained in the conference manager modules. If there is a request, the conference manager will first judge whether it is acceptable, and then inform the controller to take the corresponding actions.

Each terminal can support SVC encoding and decoding. After being encoded, the layered video data was encapsulated as real-time transport protocol (RTP)/UDP packets. We used different UDP ports to transmit these packets so that OpenFlow switches could identify the flow of a specific layer and execute the corresponding actions. AVC/SVC library decoding tools were provided to enable the SVC player to decode SVC streams and display the SVC video. There were three types of terminals in the experiment, i.e., smartphones, notebooks, and PCs with high-definition display. According to our design, these terminals with different capabilities would receive different numbers of SVC layers.

4.2 Test for bandwidth usage and delay

We evaluated the performance of bandwidth usage in the whole network and average end-to-end delay of the proposed SDN-enabled SVC multicast conferencing architecture. We compared it with the MCU-based system and SDN-enabled single-layer video multicast conferencing system, respectively. In the MCU-based system, video from the source will first be transmitted to the MCU and then forwarded to each receiver terminal. In an SDN-enabled single-layer or SVC video multicast conferencing system, the video stream will be forwarded to the receivers in a multicast manner. The major difference between these two systems is that in a single-layer video multicast system, all receivers obtain the source video at the same bit rate. However, in a layered video multicast conferencing system, devices with different capabilities can obtain the video at different bit rates. We designed an experiment with up to 50 users and implemented the same network topology and link settings in different systems. These 50 terminals requested for the same video conference from the conference manager one by one.

Fig. 6a shows the usage of network's bandwidth along with the number of terminals. As the number of participants increases, the SDN-enabled SVC multicast architecture outperforms the other two conferencing solutions in bandwidth saving. This is because, in the system with the MCU-based solution, terminals send videos to the MCU, and then the MCU sends the video back to all receivers. In an SDN-enabled single-layer multicast system, small screen devices receive videos at the same bit rate as the ones with high-definition displays. The other two conferencing solutions will increase unnecessary traffic and waste network bandwidth resource.

Fig. 6b shows the average end-to-end delay as the number of terminals increases in different systems. The performance of the SDN-enabled layered video multicast solution is almost the same as that of the SDN-enabled single-layer multicast conferencing system. Both perform better than the MCU-based solution. This is reasonable, because the video will first be transmitted to the centralized MCU and then forwarded to receivers in the MCU-based solution. It will result in larger end-to-end delay than SDN-enabled multicast solutions.

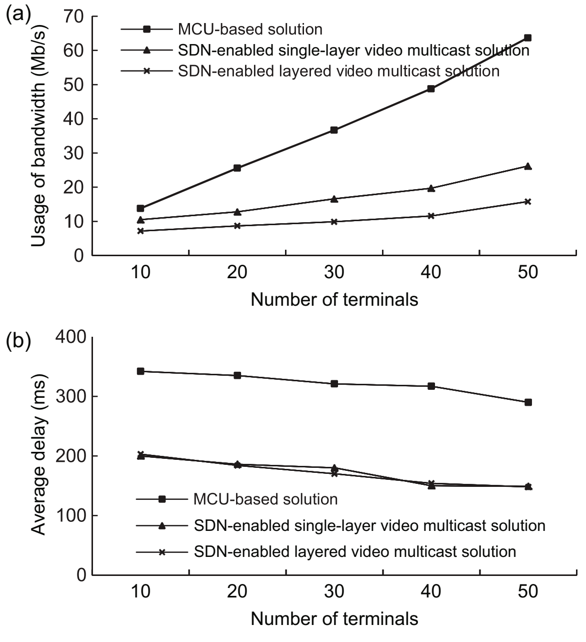

Fig. 6 Test results of bandwidth usage in the whole network (a) and average delay (b) as the number of terminals increases

4.3 Experiment for in-network adaptation

In this experiment, we evaluated the function of SVC in-network layer adaptation. We encoded a video in the source terminal into three layers of quality scalability, including a base layer, enhancement layer 1, and enhancement layer 2, and encoded the same video into the single-layer format with the same bit rate. We chose two member terminals as the receivers to compare the performance of the proposed SDN-enabled SVC multicast with that of the SDN-enabled single-layer multicast solution. To imitate network congestion, a third-party software called Ostinato (Srivats, 2010) was used to change the network traffic continuously in the system, which shared the same physical link as the selected terminals' multicast path. Then we monitored the variation of the network traffic and compared the video quality of two receivers to judge whether the in-network adaptation strategy of SDN-enabled SVC multicast works better than the single-layer video multicast conferencing system.

We measured the packet-loss ratio of the SDN-enabled layered video multicast strategy and the single-layer video multicast strategy, respectively. The results are plotted in Fig. 7. The packet-loss ratio of layered video multicast is much lower than that of the single-layer solution. The reason is that when the link becomes congested, the SDN-enabled layered video multicast conferencing system can decrease the number of forwarded layers to avoid packet loss.

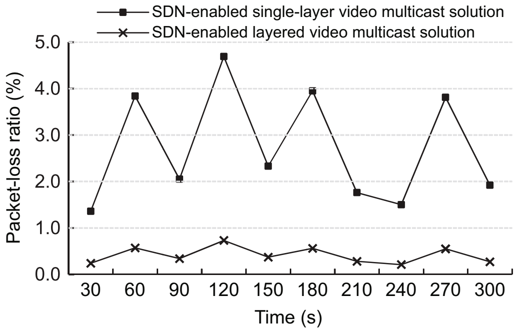

Fig. 7 Packet-loss ratio of the SDN-enabled SVC video multicast solution and the single-layer video multicast strategy

Fig. 8 shows the experimental results in terms of intuitional time-domain distribution of blurred pictures for the SDN-enabled SVC video and single-layer video multicast conferencing solutions. Packet loss may lead to blurred pictures. Normally, seriously or moderately blurred pictures will heavily degrade the conference members' QoE, while the effect of mildly blurred pictures is negligible. When the background traffic is high enough to congest the link, the SDN-enabled layered video multicast conferencing system can decrease the number of layers and achieve much fewer blurred pictures than the other solution. When the link obtains enough bandwidth to transmit the enhancement layers, it will increase the number of forwarded layers to provide a high-quality video. Hence, the in-network adaptation may substantially improve the conference members' QoE in the scenario of time-varying traffic.

The experimental results indicated that SDN-enabled SVC multicast works better than the other solution. When the network traffic changes, the SDN-enabled SVC multicast solution can

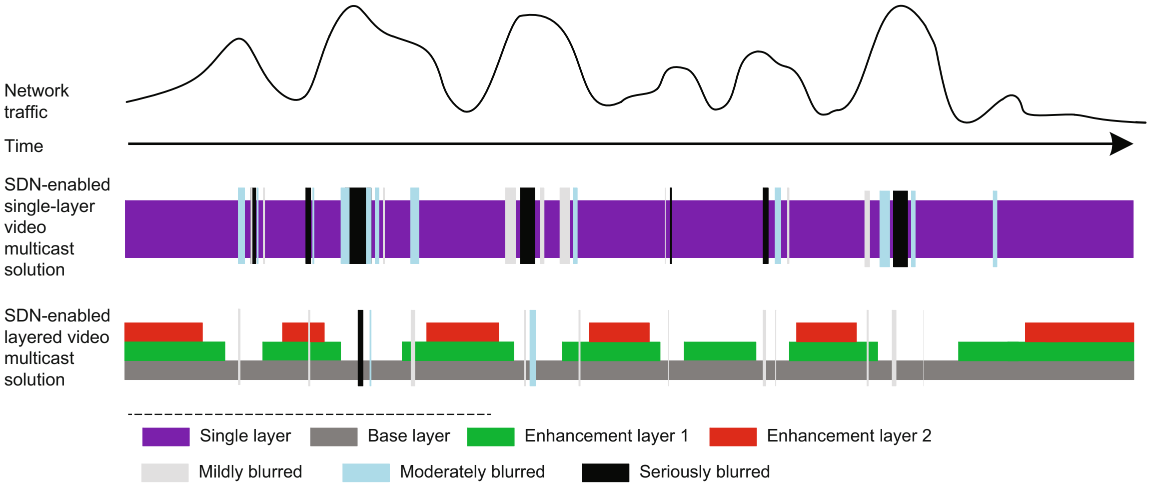

Fig. 8 Time-domain distribution of blurred pictures of the SDN-enabled SVC video multicast solution and the SDN-enabled single-layer video multicast conferencing system (References to color refer to the online version of this figure)

dynamically adjust the number of video layers to obtain a better video quality. The flexible and controllable SVC multicast stream service can obviously reduce the packet-loss ratio and try its best to improve the quality of conference video, guaranteeing fluency and stability to satisfy the conference members.

5 Conclusions

In this paper, we presented the design and deployment of a video conferencing system based on SDN-enabled SVC multicasting, which flexibly customizes multicast paths according to the network state. It also supports heterogeneous terminal devices and in-network transmission adaptation. Experimental results were presented, showing that the proposed system can not only provide a flexible and controllable video delivery but also reduce the network bandwidth usage and guarantee the quality of a video conference.

References

Avaya, 2016. Scalable Video Coding Algorithm. Available from http://docs.radvision.com/ [Accessed on Feb. 25, 2016].

Civanlar, S., Parlakisik, M., Tekalp, A.M., *et al.*, 2010. A QoS-enabled OpenFlow environment for scalable video streaming. IEEE Globecom Workshops, p.351-356. http://dx.doi.org/10.1109/GLOCOMW.2010.5700340

Drutskoy, D., Keller, E., Rexford, J., 2013. Scalable network virtualization in software-defined networks. *IEEE Internet Comput.*, **17**(2):20-27. http://dx.doi.org/10.1109/MIC.2012.144

Egilmez, H.E., Civanlar, S., Tekalp, A.M., 2013. An optimization framework for QoS-enabled adaptive video streaming over OpenFlow networks. *IEEE Trans. Multim.*, **15**(3):710-715. http://dx.doi.org/10.1109/TMM.2012.2232645

Feng, H.E., Wu, W.J., 2012. Framework and user migration strategy of cloud-based video conference multi-gateway system. 19th Int. Conf. on High Performance Computing, p.1-8. http://dx.doi.org/10.1109/HiPC.2012.6507521

Fenner, W.C., 1997. Internet Group Management Protocol, Version 2, FRC 2336. Internet Engineering Task Force, Fremont. http://dx.doi.org/10.17487/RFC2236

Li, X., Ammar, M.H., Paul, S., 1999. Video multicast over the Internet. *IEEE Netw.*, **13**(2):46-60. http://dx.doi.org/10.1109/65.768488

McCanne, S., Jacobson, V., Vetterli, M., 1996. Receiver-driven layered multicast. *ACM SIGCOMM Comput. Commun. Rev.*, **26**(4):117-130. http://dx.doi.org/10.1145/248157.248168

McKeown, N., Anderson, T., Balakrishnan, H., *et al.*, 2008. OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM Comput. Commun. Rev.*, **38**(2):69-74. http://dx.doi.org/10.1145/1355734.1355746

Ng, K.F., Ching, M.Y., Liu, Y., *et al.*, 2014. A P2P-MCU approach to multi-party video conference with WebRTC. *Int. J. Fut. Comput. Commun.*, **3**(5):319-324. http://dx.doi.org/10.7763/IJFCC.2014.V3.319

Open Networking Lab, 2015. POX Wiki. Available from http://openflow.stanford.edu/display/ONL/POX+Wiki [Accessed on Sep. 22, 2015].

OpenWrt Developer Team, 2015. OpenWrt Project. Available from https://www.openwrt.org/ [Accessed on Aug. 15, 2015].

Polycom, 2014. SVC-Based Conferencing Solutions Deployment Guide. Available from http://support.polycom.com/global/documents/support/user/products/network/RMX_SVC_Deployment_Guide_V8_3.pdf [Accessed on Jan. 18, 2016].

Qiu, R.B., Kuhns, F., Cox, J.R., 2002. A conference control protocol for highly interactive video-conferencing. IEEE Global Telecommunications Conf., p.2021-2025. http://dx.doi.org/10.1109/GLOCOM.2002.1188554

Schwarz, H., Marpe, D., Wiegand, T., 2007. Overview of the scalable video coding extension of the H.264/AVC standard. *IEEE Trans. Circ. Syst. Video Technol.*, **17**(9):1103-1120. http://dx.doi.org/10.1109/TCSVT.2007.905532

Sezer, S., Scott-Hayward, S., Chouhan, P.K., *et al.*, 2013. Are we ready for SDN? Implementation challenges for software-defined networks. *IEEE Commun. Mag.*, **51**(7):36-43. http:/dx.doi.org/10.1109/MCOM.2013.6553676

Srivats, P., 2010. Ostinato—Network Traffic Generator and Analyzer. Available from http://www.ostinato.org/ [Accessed on Jan. 12, 2015].

Willebeek-LeMair, M.H., Kandlur, D.D., Shae, Z.Y., 1994. On multipoint control units for videoconferencing. Proc. 19th Conf. on Local Computer Networks, p.356-364. http://dx.doi.org/10.1109/LCN.1994.386585

Zhang, S.P., Niu, D., Hu, Y.C., *et al.*, 2014. Server selection and topology control for multi-party video conferences. Proc. Network and Operating System Support on Digital Audio and Video Workshop, p.43-48. http://dx.doi.org/10.1145/2578260.2578261

Zhang, Z.C., Li, V.O.K., 2002. Router-assisted layered multicast. IEEE Int. Conf. on Communications, p.2657-2661. http://dx.doi.org/10.1109/ICC.2002.997323

Zhao, M., Jia, B., Wu, M.Q., *et al.*, 2014. Software defined network-enabled multicast for multi-party video conferencing systems. IEEE Int. Conf. on Communications, p.1729-1735. http://dx.doi.org/10.1109/ICC.2014.6883572