IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 29, 2019, accepted December 17, 2019, date of publication January 7, 2020, date of current version January 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2964630

Software-Based Video–Audio Production Mixer via an IP Network

DARIO LUZURIAGA, CHUNG-HORNG LUNG, AND MARGARET FUNMILAYO
Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada
Corresponding authors: Dario Luzuriaga (dario.luzuriaga@carleton.ca) and Chung-Horng Lung (chlung@sce.carleton.ca)

ABSTRACT Modern television production has promoted the simultaneous use of several cameras and sound sources, which increases the complexity and costs of broadcasting studios. This paper describes the design and implementation of a video-audio production mixer via an IP network. It is presented as a potential replacement for traditional professional production systems on the basis of cost reduction, as it is a software-based system that uses the existing technologies and can be built in community television stations and economic private productions. A prototype combining five cameras, a title generator, a multimedia player, microphone sound, music, and other resources for video recording or Internet live streaming has been implemented. The system also features a voice intercommunication capability to support teamwork. This technology has been mounted using different high-definition cameras with High-Definition Multimedia Interface (HDMI) outputs, desktop computers, mobile phones and other non-dedicated equipment available for free or at a low cost. Although it works on non-dedicated hardware, this system provides video routing, sound managing, and audiovisual mixing with an approximate total delay of only 1.4 s. It has been mounted mostly on Linux environments to guarantee reliability and the extensive use of free software, which demonstrates the feasibility of building a cost-effective video-audio production mixer, using the available devices and techniques.

INDEX TERMS Audio, multicast, network, routing, television, video.

I. INTRODUCTION

Television production may differ from its immediate historical predecessor, filmmaking, in the concept of live content generation. While modern filmmaking puts a considerable amount of effort into mounting celluloid and sound support, or its equivalent video editing, television has resorted to a combination of resources to produce content with a ''live'' style, based on the need to broadcast several hours of programs per day. Given this context, instead of using a single camera as in traditional filmmaking, television production has promoted the simultaneous use of several cameras and several sound sources, which increases the complexity of broadcasting studios. In addition, although technological advances may simplify the use of hardware today [1], the resources required for live productions may still hinder some content generation projects [2], especially those with a low budget. To respond to this widespread challenge, our goal was to design more affordable live multicamera mixing technologies, by using or adapting open-source software and commercial off-the-shelf (COTS) technologies to ensure cost-effectiveness.

In particular, the objective of this research project was to set up a software-based video-audio mixer to support live television production formats. The entire system consists of multiple components, including an open-source mixer application called Open Broadcaster Software (OBS) Studio [3], which is a free software. Camera sources, titles, video playback, microphone sound, and audio signals are converted into IP packets, which are sent over a reliable high-speed network. The mixer is also capable of visually synchronizing actions in cameras and microphones, so that movements look continuous from all points of view used. Sound in this system was mixed separately via a set of tools known as Linux KX Studio [4]. Then, the sound mix was sent to OBS Studio. Using a smartphone application (Unity Intercom), IP voice internal communications were integrated by means of Wi-Fi, another built-in capability.

This article describes how different applications have been designed and mounted to interact with the hardware used. To illustrate this presentation, some audiovisual materials help describe our video-audio production mixer

The associate editor coordinating the review of this manuscript and approving it for publication was Jose Saldana.

and reinforce some of the concepts in this article. These materials are available at: https://youtu.be/eZ8-Y0MjkHE or https://youtu.be/BPne3GmZcm4 (with English subtitles).[1]

The rest of the paper is structured as follows: a description of the context of our research is presented in Section II-A; our research assumptions on quality of experience are described in Section II-B; a high-level approach to multicamera setups is presented in Section III-A; considerations on advantages with respect to the cost of this system are discussed in Section III-B; an introduction to the technical configuration steps of the system is provided in Section IV, while those steps are developed in Sections IV-A to IV-G. Finally, our conclusions and final remarks are summarized in Section V.

II. CONTEXT OF THE RESEARCH

This paper has been conceived to use techniques in communications engineering and system tools to solve concrete problems in audiovisual production. As a result, its utility is based on existing techniques and applications, rather than on abstract speculations. Before exploring our solution for low-cost camera and sound mixing, the state of the art is presented in Section II-A. Considerations on quality of experience (QoE) are later discussed in Section II-B.

A. STATE OF THE ART

In recent time, television has been changing from a technological point of view. There have been new developments in transmitting content digitally [5] and even making it more interactive [6] (gaming, content on demand, etc.). However, regardless of the new formats for audiovisual distribution, mixing video feeds from live cameras and sounds will probably remain an essential need in future television in terms of content generation.

The first challenge community stations and low-budget productions face is the high costs of multicamera and multisound productions.[2] They require several technological and human resources to "narrate with images and sounds."

For example, the British Broadcasting Corporation (BBC) has developed as of 2014 a multicamera video mixer to combine "nearly live" camera sources [7] and streaming over the Internet. Briefly, the project uses ultra-high definition (UHD) static cameras capable of capturing video in 3840 × 2160 pixels (pxs), also known as 4K, so that a single operator can reframe, crop and zoom in/out parts of a video while mixing feeds from different cameras. To complete all these tasks, the operator has to prepare work in advance for the mix to resemble a live mix. This requires buffer time, which depends on the operators, their experience and the complexity of the production. For this BBC project in particular, the sound has been obtained from the event's sound feed, which is usually provided by the event's organizers (like in music concerts).

Furthermore, the sound mixing required for on-site productions during public performances may differ from sound in broadcasting (ambient sound, levels of drums sound, etc.). Therefore, sound requires special treatment and mixing in multicasting, as it has been considered in this paper.[3]

A drawback of the approach presented in Sections II-A to II-G is that, to operate the video-audio mixer under current conditions, more human resources, including sound operators or camera operators, are required. Nevertheless, a future line of research in our project may consider adding a dedicated artificial intelligence (AI) system to operate cameras or determine how relevant audio sources are. For example, this system could understand and execute voice instructions, and work autonomously.[4]

B. CONSIDERATIONS ON QUALITY OF EXPERIENCE

Research efforts are often based on certain assumptions and even in rigorous logic systems [8], the role of these assumptions is vital to increase knowledge [9]. In this paper, assumptions were mostly related to quality of experience (QoE).

QoE is a set of opinions provided by technology users who rate their experience with those technologies. This concept refers to user satisfaction. In addition, there is the concept of quality of service (QoS), which covers a set of specifications provided by technology and application manufacturers about the quality of their products or technical services [10].

QoE for multimedia products usually requires large user surveys carried out in laboratories or in those environments where users operate the technologies in question [11]. Also, these surveys must take into consideration the different variables of networks that users may usually have access to, including the complexity of wireless technologies [12]. However, these surveys were outside the scope and budget of this paper. As a result, although it is clear that other researchers could benefit from QoE surveys on systems such as the one we describe here, our research has not been designed to include QoE tests, since a thorough QoE evaluation would involve multiple users in various scenarios. In addition, QoE is highly related to delay, jitter and video/voice synchronization, all issues that have been addressed in the design of the study itself. Some QoE guidelines have been taken into account, though no social considerations have been included.

This paper is based on at least two types of QoE assumptions: assumptions about the audience's experience

[1] Alternate links to these materials are: https://vimeo.com/382509602 and https://vimeo.com/382506881.

[2] In this paper, the expression "multicamera" is usually used to suggest that feeds from several camera sources are mixed, whereas the expression "multisound" indicates that feeds from several audio sources are combined. The first concept will be used for video mixing capabilities, whereas the second will refer to audio mixing. The whole system explored in this paper is considered a multicamera and multisound technology.

[3] An example are drums. Depending on the size of a stage, spectators on the site might hear the sound of the drums directly from the instrument, and thus no special sound mixing will be required. However, in multicasting, this sound source must be mixed. The same principle applies to ambient sound when, for example, an audience sings as one. This sound source is needed for multicasting, but it is not required on the speakers to provide sound to those spectators near the stage. This paper has taken into consideration this aspect. In contrast, it seems that the BBC project mainly uses mixing to obtain this sound for the on-site spectators.

[4] This idea, which has not been developed in this paper, could help resemble the relationship between a TV director operating a video mixer and his or her team of technicians, who follow his or her oral instructions.

TABLE 1. QoE assumptions for spectators.

Spectators' experience	QoE assumption
Acceptable video recording quality	OBS Studio maximum quality setup: 1920 × 1080 pxs at 30 fps, MPEG2 TS compression, Lossless Quality output mode, very large files (around 3.73 GB/min)
Acceptable audio recording quality	OBS Studio 44 KHz CD quality audio, MPEG2
Acceptable live streaming quality	OBS Studio YouTube streaming provider pre-set 2019

QoE: Quality of Experience

(users watching and listening to audiovisual materials with certain *compression*, frame rate and video quality) and assumptions about producers' experience (producers of audiovisual content operating the video and sound mixer for live streaming or recording). The audience's relevant questions could include the following: *Is the picture clear enough? Is the picture synchronized with the voices? Do movements look robot-like? Does the sound have too much noise? Is the delay acceptable for the actions on the stage to match a projection on a big screen?* In contrast, questions for producers could include the following: *Is the system delay acceptable for watching camera movements fast enough, while camera operators follow the director's instructions? Do actions look synchronized on a projection screen near the stage before mounting a set?*

With respect to the first group of assumptions, for which other approaches could use QoE surveys, Table 1 shows some technical details on visualization from the point of view of spectators.

The assumptions listed on Table 1 could change depending on applications and distribution media. These data are mostly based on community broadcast production flow, where highest quality materials might be required for broadcasting or storage. YouTube settings for live streaming are an assumption here since they are one of the most popular.

With regard to the producers of audiovisual content, their assessment of the standards of production flow is also subjective. These standards include the maximum number of cameras in the video mixer and the total system delay, which is defined as the time it takes for an action before the cameras and microphones to produce an audiovisual representation in the output of the system. The output could be a projection on a screen or on the screens installed where the video and sound mixer is located (a main assumption in this paper).

Table 2 shows some of the QoE assumptions made regarding audiovisual content producers, which are based on personal experiences. In the case of a projection on a big screen located close to an event's stage, some of the most noticeable delays are seen between lip movement and the actions that are represented late on a big screen. This is another subjective evaluation. In a research project based on social surveys, this evaluation could be supplemented by QoE inquiries, taking

TABLE 2. QoE assumptions for producers.

Producers' experience	QoE assumption
Maximum number of cameras for video mixing	5
System delay for a projection on a screen near the event's stage	Less than 1 s
System delay for recording or live streaming without a projection near the stage	Less than 3 s

into consideration the perspectives of both producers and potential spectators. In addition, the less-than-one-second delay mentioned in this paper is an assumption based mostly on the professional experiences of audiovisual producers in the field.

Without a projection screen for the spectators located close to the stage, there are practical and operating criteria that help define assumptions on acceptable system delays. A director operating a video mixer might request that camera operators perform certain camera movements. Those operators will need time to understand the director's oral instructions and execute them. If the system delay is significantly longer (for example, 15 s), it will be very difficult for someone to operate the mixing system. Again, assuming that the total system delay is of less than 3 s is another subjective evaluation based on individual professional experiences.

III. DESIGN OF THE PRODUCTION SYSTEM

The design of the system covered some typical features required for low-cost multicamera productions.

A. PRODUCTION SETUP

Multicamera productions may have different setups: fixed configurations in permanent or main studios or temporary designs for special events or at specific locations. Fig. 1 shows a potential setup to account for multiple production alternatives.

In this example, two areas can be differentiated: the *production* area and the *control* area. The production area (sometimes called *plateau*) is the part of a setup where actions take place: events, performances, announcements, etc. Therefore, cameras, microphones, and lights are located there, controlled by camera operators, sound technicians, the lighting crew, etc.

The control area includes the equipment and human resources responsible for combining the audiovisual signals coming from the *plateau*, where the video mixer is located and controlled by a content director. Mixing sound usually requires a separate audio mixer, which is operated by a sound technician. Also, titles on the screen require dedicated treatment with a separate title generator and, sometimes, a special operator. To simplify the configuration, the title generator and the sound mixer are mounted on the same equipment, but this may not be the case in some complex productions. The program producer sometimes works in the control area too.

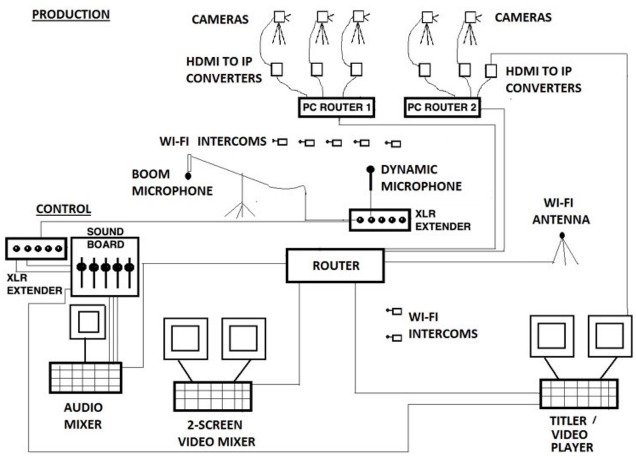

FIGURE 1. **Audiovisual production setup.**

Some kind of voice intercommunication system is usually necessary between the production and the control areas. For example, the camera operators receive instructions from the director, as it has been mentioned before. There might be direction assistants and production assistants that also need to communicate with the director and the producer, respectively.

Table 3 shows key equipment used to design our video-audio mixer. Some devices are directly linked to Fig. 1 and are thus shown in italics. Other devices are ignored since they are considered generic (like mobile Android or iOS devices used as voice *intercoms*).

For the first experiments, only one PC router was used; however, later on, PC router 1 and PC router 2 were added to the setup to improve performance. Almost all the practical applications of the devices in Table 3 will be described in the coming sections.

With regard to physical distribution, the PC routers could be located in the *plateau* area if the production and the control areas were separated by a considerable distance. This could help save wire. The HDMI to IP converters, which are also located in the *plateau*, generate video signals in UDP packets for network distribution (see Section IV-A).

B. COST-EFFECTIVENESS

For reference purposes only, Table 3 includes a column with the approximate prices of most of the equipment used in this research project. The price of each item is the actual price spent on the device or the price posted by the seller when the experiments were carried out. The total amounts to CAN$ 5437.

[5]This microphone is also known as a dynamic microphone, but it could be used as a boom microphone by adding amplification in small-budget productions. A more appropriate boom microphone would be a Sennheiser MKE 600, for example. A boom microphone is a microphone mounted on a large extendable arm to locate it above the performers, out of the camera frame. It is usually a very sensitive microphone.

TABLE 3. **Main hardware required to set up the low-cost video-audio mixer.**

Quantity	Equipment	Function	Cost (CAN$)
2	Canon Vixia HF G20 *camera*	Capturing visual actions	1200
1	JVC GZ-E200 *camera*	Capturing visual actions	100
1	JVC GZ-HM440RU *camera*	Capturing visual actions	100
1	ION Air Pro Lite action *camera*	Capturing visual actions	54
6	eSYNIC HDMI to LAN transmitter	Converting video to IP	223
6	eSYNIC HDMI to LAN receiver	Converting IP to video for monitoring purposes only	223
1	D-Link DSR-250 gigabit router	Routing everything except video feeds from cameras	157
1	D-Link DGS-1100 24-port gigabit managed switch	Linking camera transmitters and computers on VLANs	204
2	D-Link GO-SW-8G gigabit switch	Linking computers to router	68
1	D-Link AC1750 Wi-Fi gigabit *router*	Connecting phones to Unity server	136
9	Trendnet TU3-ETG USB 3.0 to gigabit LAN adapter	Connecting servers to the VLAN of each camera	204
1	Eurorack UB1204FX-Pro *soundboard*	Connecting microphones with XLR to miniplug PCs	84
1	Numark DM-3000X soundboard mixer	Amplifying sound	55
1	BobCat LAN sound extender transmitter	Extending XLR microphones with LAN wires	100
1	BobCat LAN sound extender receiver	Extending XLR microphones with LAN wires	100

TABLE 3. *(Continued.)* **Main hardware required to set up the low-cost video-audio mixer.**

1	Sennheiser E815S (boom) microphone[5]	Capturing voices and sounds	67
1	Sennheiser E935 (dynamic) microphone	Capturing voices and sounds	54
1	*Router 1* Dell Vostro230 8-GB RAM Debian *PC*	Routing 3 cameras on VLANs	150
1	*Router 2* Optiplex 760 4-GB RAM Debian *PC*	Routing 2 cameras and titler on VLANs	80
1	Lenovo ThinkCentre M58P PC, KX Studio *sound mixer*, and OpenLP *titler*	Adding titles, playback and mixing sound	100
1	HP CE8300 10GB RAM computer, OBS Studio Ubuntu *video mixer*	Mixing video, receiving sound, recording and streaming	160
1	Technics SAGX130 sound amplifier	Amplifying sound for OBS	84
2	Yamaha NS-A636140-W speakers	Sound output for OBS Studio monitoring	134
1	Macbook Pro 13-in Retina Unity server	Linking mobile phones via Wi-Fi to provide voice intercoms	1600
		Total:	5437

Nevertheless, we already had many of the devices required for the video-audio mixer, and we can assume that community television stations also own these devices. The added value of this research is providing an alternative to adapt existing equipment to a flexible setup. The dedicated devices included in our research are mostly related to video mixing: eSYNIC High Definition Multimedia Interface (HDMI) to Local Area Network (LAN).[6] extenders, D-Link router and switches, Trendnet TU3-ETG Universal Serial Bus (USB) to LAN adapters. The total price of all of them amounts to CAN$ 1215. Computers with other uses or regular/personal computers can be adapted for an estimated total of around CAN$ 500. The estimated cost of the video mixing equipment is CAN$ 1705.

In contrast, at the time we carried out this research, the switcher Ross Carbonite Black Solo all-in-one [13], one of the least expensive video mixers, which is manufactured by Canadian company Ross Video, cost CAN$ 12,110.[7] This mixer features three HDMI and six Serial Digital Interface (SDI).[8] video inputs. As a result, to have a total of six HDMI inputs (for cameras with the same connection), three SDI 3G to HDMI converters are required, such as an eSYNIC SDI-HDMI converter, which add up to a total of CAN$ 166 (three devices).

Adapting non-dedicated equipment to a flexible video mixing system is clearly more economic in terms of hardware (only 14% of the cost of a commercial alternative),.[9] but it demands more time and effort spent on configuration.

With regard to the sound mixing equipment, its cost can probably be kept similar to that of the video mixing system, considering the fact that the Eurorack UB1204FX-Pro soundboard (or any other device of the same kind) can directly mix sound and send it to one of the streaming or synchronization mechanisms offered by commercial video mixers.

IV. DESIGN AND IMPLEMENTATION STEPS

This section covers the general steps that were followed to implement the video-audio production mixer.

These general steps are as follows:

- **Network design and configuration** (Sections IV-A and IV-B): This is a general approach, since subnetworks can be set up differently depending on the routers and switches used.
- **Video OBS Studio design and configuration** (Section IV-C): A global design is provided to define a multicamera mixing tool.
- **Sound KX Studio configuration** (Section IV-D): The basic principles of sound virtual connections, as well as the methods to generate a virtual audio mixer, are explained.
- **Media source synchronization** (Section IV-E): A method to synchronize cameras and sound sources based on the matching of visual actions is presented.
- **Multitrack audiovisual recording setup** (Section IV-F): Sound recording using the application Ardour is described. Also, adapting the video server/s to record feeds from camera and video sources individually is suggested. These methods will help users perform remixing via editing (or *montage*) later.
- **Voice intercommunication** (Section IV-G): Use of the commercial application Unity Intercom to turn mobile phones into intercommunication devices is described.

[6] The device mentioned here (HDMI to LAN extender) is presented in Fig. 2. It converts multimedia signals (sound and/or video) into IP computer network signals.

[7] This video mixer presents more features other than mixing camera or video sources; however, it is still a suitable alternative available in Canada for low cost productions, as it was in part conceived for educational activities, corporate settings, and religious services.

[8] This connector transmits video using coaxial cables.

[9] CAN$ 12,110 ________________ 100%
CAN$ 1705 ________________ 1705 x 100%/12 110 ≈ 14.08%

FIGURE 2. **HDMI to IP converter (video transmitter).**

- **Tests** (Section IV-H): Some system saturation tests were carried out to determine for how long the system could run properly without noticeable negative results.

A. NETWORK DESIGN

The video mixer has been designed to work with any high definition (HD) cameras with HDMI output available in the market. HD cameras are known for the size of their visual frames (1920 × 1080 pxs). Experiments were carried out using cameras with a rate of 30 frames per second (fps), as shown in Table 3.

The camera outputs are sent to COTS devices, which are known as HDMI to IP converters. They are basically transmitters[10] of MPEG2-encoded video obtained from HDMI sources and converted into User Datagram Protocol (UDP)[11] packets for multicasting. Fig. 2 shows one of these devices, manufactured by eSYNIC.

This transmitter was designed to multicast video, *i.e.*, multiple receivers can obtain the video signal from the same source at the same time. Multicast is a complex technique in communications engineering that has a relatively long history in multimedia [14], several developments [15], [16], and new applications [17], to allow networks to transmit data effectively without congestions [18]. In our research, it has not been necessary to focus on multicast effectiveness since traffic is limited to six sources and one receiver, as it will be described later.

The multicast receiver is a device similar on the outside to the transmitter displayed in Figure 2. It has been connected to a video screen via its HDMI output. Receiver and transmitter were initially connected to each other via a LAN cable (CAT5 or CAT6[12]), so that the receiver could obtain an UDP video and convert it into an HDMI signal.

An unmanaged D-Link GO-SW-8G switch was then placed between one of the transmitters and the receiver, so that a computer could also be connected to the switch to study the UDP traffic using *WireShark* [19]. Based on the results obtained, video from the source camera was received using the Linux command **vlc udp://@239.255.42.42:5004**, via a VLC Player [20], displaying a high definition equivalent picture at a 30-fps rate.

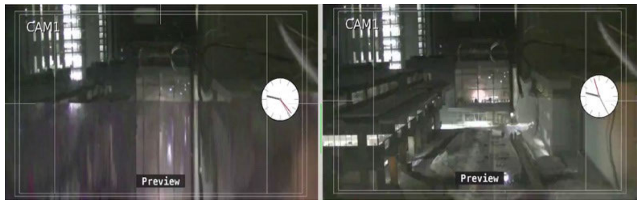

FIGURE 3. **Images received when the network included different numbers of transmitters. (a) Two transmitters. (b) One transmitter.**

Although the receiver and one of the transmitters were able to communicate clear high-definition images, interference was detected on the side of the receiver when using more than one transmitter. To keep a clear reception, a single transmitter was left in the network.

Figure 3(a) illustrates interference resulting from two video transmitters in the same network, whereas Figure 3(b) shows clear reception after disconnecting the second transmitter and leaving only one transmitter in the network.[13]

These findings suggested the need to isolate signals from multiple transmitters. With this aim in mind, artificial subnetworks were designed using dedicated software. In this paper, subnetworks are referred as virtual LANs (VLANs) for routing purposes [21]. VLANs are divisions of the network made using software to isolate multimedia traffic.

Figure 4 shows a high-level approach to the network we have built to solve video interference produced when using more than one camera transmitter within the same network. Some devices, including managed switches or receivers, have been ignored at this stage for didactical reasons.

A set of five cameras (CAM1 to CAM5) is connected to video transmitters (HDMI to IP converters). CAM6 is the video computer output of the OpenLP title generator and uses another HDMI to IP transmitter. Each video transmitter is connected to a 24-port D-Link DGS-1100 gigabit-managed switch [22], which is located before the PC router by means of regular LAN cables.[14] This switch was set up to isolate the traffic of each transmitter by building six different virtual LANs: VLAN11 to VLAN16. It could be accessed via IP address 1.1.0.1.

A special video server computer (*PC router* in Fig. 4) has been included in the network design. It is a Dell Vostro 230 computer with an 8-GB RAM and Linux Debian 9 operating

[10] Video transmitters can send HD video feeds at a maximum rate of 60 fps. Cameras need half of such a rate.

[11] It is a communication protocol for computer networks mostly used to transmit video, sound, and multimedia data. It is effective for live broadcasting.

[12] Both CAT5 and CAT6 cables work with LAN connectors, but differ in the amount of information they can transport simultaneously. Even the minimum speed offered by CAT5 cables (100 Mbps) is enough to communicate the signal of around 17 Mbps generated in the transmitter. However, in a subsequent stage during network design, we decided that it was more convenient to use CAT6 cables since they provide up to 10 Gbps in a 100-m length, and data congestion is never a problem.

[13] The pictures used here were obtained from OBS Studio's interface, not VLC Player. Nevertheless, results were similar when using VLC Player as a receiver software.

[14] The DGS-1100-24 managed switch placed in front of PC router is not shown in Fig. 4 to avoid confusion. This scheme is a simplified representation of the system. In fact, the DGS-1100-24 switch could have been avoided at this step. It helps organize, label, and rewire connections during tests or configuration.

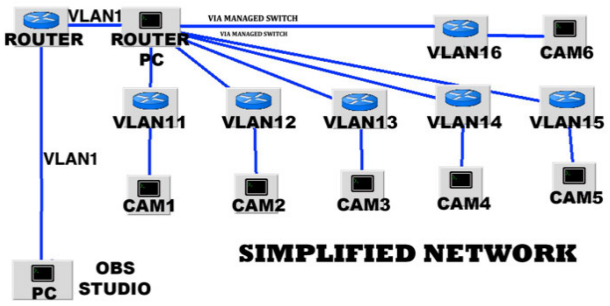

FIGURE 4. Topology of VLANs used to remove interferences.

system. This server has a main gigabit Ethernet interface connected to a D-Link DSR-250 gigabit router [23] (*router* in Fig. 4) within a new virtual network: VLAN1. The PC router also includes six external Trendnet TU3-ETG USB 3.0 to Ethernet gigabit interfaces, so that the server computer can be connected individually to the six virtual LANs.

Finally, an HP CE8300 computer with a 10-GB RAM and Linux Ubuntu 18 operating system has been configured with OBS Studio 22 (*OBS Studio* in Fig. 4). It is connected to the VLAN1 of router DSR-250 and assigns a specific IP address (1.0.1.2) to it. VLAN1 also enables Internet connection so that applications and streaming materials can be installed outside of the network.

The network design in Fig. 4 is for reference purposes only. This topology becomes more complex when additional features are included in the system.

B. ROUTING VIDEO

The server computer Vostro 230 (*PC router* in Fig. 4) has to be completely isolated from the video traffic in each VLAN. The solution for this is to create six different virtual machines running on Linux Debian 9. The application Virtual Box [24] for Linux has been chosen for creating the virtual machines. By default, Virtual Box can only manage four LAN interfaces. Since in this case seven LAN interfaces are needed, a special setup via Linux commands is necessary for Virtual Box to recognize all the seven LAN interfaces.[15] Virtual Box needs to be configured in a way that the virtual machines are connected to both VLAN1 and one of the VLANs of the video transmitters, as shown in Table 4.

Each of the six virtual machines required special treatment and configuration. Once this step was completed, the PC router capability to forward videos needed to be implemented using an application known as Socat [25]. Socat is a program designed to forward network traffic to specific destinations, multicasting or broadcasting.[16] It also intercepts

[15]On the next pages, it will become evident that this special configuration is not required. A second PC router has been added to improve performance, as anticipated.

[16]Broadcasting is a type of multicasting. However, while content might be shared with some receivers in multicasting communication, content is shared with all the available receivers in broadcasting communication. Throughout this paper, *multicast* and *broadcast* can be considered synonyms.

TABLE 4. Access to VLANs through virtual machines of the PC router.

Virtual machine	VLAN access
CAM1_VM	VLAN1, VLAN11
CAM2_VM	VLAN1, VLAN12
CAM3_VM	VLAN1, VLAN13
CAM4_VM	VLAN1, VLAN14
CAM5_VM	VLAN1, VLAN15
CAM6_VM	VLAN1, VLAN16

TABLE 5. Commands to forward video from PC router to OBS.

Virtual machine	Socat command
CAM1_VM	socat udp4-recv:5004,bind=239.255.42.42,ip-add-membership=239.255.42.42:enp0s1, reuseaddr udp4-sendto:1.0.1.2:10001
CAM2_VM	socat udp4-recv:5004,bind=239.255.42.42,ip-add-membership=239.255.42.42:enp0s2, reuseaddr udp4-sendto:1.0.1.2:10002
CAM3_VM	socat udp4-recv:5004,bind=239.255.42.42,ip-add-membership=239.255.42.42:enp0s3, reuseaddr udp4-sendto:1.0.1.2:10003
CAM4_VM	socat udp4-recv:5004,bind=239.255.42.42,ip-add-membership=239.255.42.42:enp0s4, reuseaddr udp4-sendto:1.0.1.2:10004
CAM5_VM	socat udp4-recv:5004,bind=239.255.42.42,ip-add-membership=239.255.42.42:enp0s5, reuseaddr udp4-sendto:1.0.1.2:10005
CAM6_VM	socat udp4-recv:5004,bind=239.255.42.42,ip-add-membership=239.255.42.42:enp0s6, reuseaddr udp4-sendto:1.0.1.2:10006

multicast traffic and sends it to specific locations.[17] A Socat command is run on each virtual machine, as presented in Table 5.

For example, the first command in Table 5 means: receive UDP video from device with IP address 239.255.42.42 and port 5004 (video transmitter) by joining its multicast membership via the *enp0s1* LAN interface and forward the packets as UDP videos to the computer with OBS Studio, identified by IP address 1.0.1.2, on its port 10001. This command corresponds to CAM1. For the other camera and video sources, the major change is related to the name of its LAN interface (enp0sX) and, above all, the port number, which increases from 10002 to 10006 for CAM2 to CAM6, respectively. Using this design, the different camera and video signals are

[17]From a Linux Windows terminal, the following commands are used to install Socat:
sudo apt-get update
sudo apt-get install socat

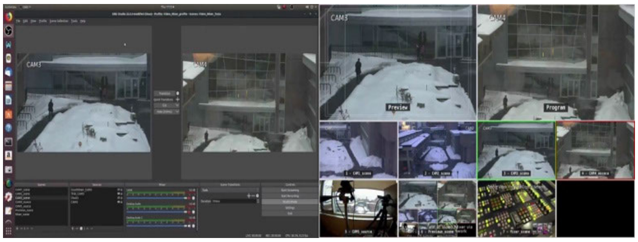

FIGURE 5. Two-screen OBS Studio interface.

basically forwarded to different ports of the same OBS Studio device, thus avoiding interference.

It is important to note that two video routers can be identified (Table 3): router 1 (Dell Vostro 230 8-GBRAM Linux PC) and router 2 (Dell Optiplex 760 4-GB RAM Linux PC). Initially, only the Vostro 230 video router was used to forward video. Now, this device forwards up to three camera signals (CAM1 to CAM3), and the Dell Optiplex 760 computer has been added to forward the three other video sources (CAM4 to CAM6). The Optiplex 760 computer has been mounted on Linux Ubuntu 18 with Virtual Box, thus providing three virtual machines on Linux Debian 9 (CAM4_VM, CAM5_VM, and CAM6_VM).

Reducing the amount of video sources, or adding a second router, helped the system perform better by decreasing up to 52 ms receiving delays on the OBS Studio computer. As a result, we have kept both PC router 1 and PC router 2 in the configuration.

C. MIXING VIDEO

In our design, the computer with OBS Studio has been adapted to create an effective video mixer. OBS Studio is mostly used in video games, tutorials, and narrations with a single camera. Figure 5 shows the two-screen interface adapted to carry out multicamera work on OBS Studio.

Each of the video sources in Figure 5 has been set up by adding a new *scene* (CAM1_scene to CAM6_scene) and a new *media source* inside each scene (CAM1 to CAM6). Each media source input is a setting *udp://@:1000x*, where the *x* is the source number. For example, for CAM1 receiving UDP packets on port 10001, the media source input is *udp://@:10001*. CAM6 has been chosen for the title generator and the OpenLP media player.[18]

The right screen in Figure 5 shows the five camera sources, CAM1 to CAM5, in small frames, plus other media sources. On top, there are two bigger frames: Preview (on the left) and Program (on the right), which emulate traditional television mixers with one preview screen and one program screen.

The preview frame is used to choose the next camera source, and to check whether the composition is adequate or needs correction. When the operator (director) is sure that the composition is right, he or she can select that source for output recording or streaming, by placing the preview content into the program frame.

[18]OpenLP is used for displaying graphs in full screen as any other video camera source. However, to add titles, a special configuration on OBS Studio is required. This includes the Superimpose function and the use of key transparencies on OBS Studio filters.

TABLE 6. OBS studio video mixing keys.

Operation	OBS keyboard key
Preview camera 1 (CAM1)	1
Preview camera 2 (CAM2)	2
Preview camera 3 (CAM3)	3
Preview camera 4 (CAM4)	4
Preview camera 5 (CAM5)	5
Preview OpenLP (CAM6)	6
Program output from preview by cut	9
Program output from preview by fade[19]	0

Traditionally, video mixers are hardware-dedicated units with several buttons for specific operations, which makes them an expensive solution. By default, the video mixer presented in this paper, which is mounted on COTS and OBS Studio, has a default interface based on mouse clicks only.

The computer keyboard has been set up to completely replace mouse clicks, so that video can be mixed by simply pressing numerical keys. Table 6 shows the operations related to different numerical keys. This setup has been configured via the OBS Studio menu "OBS"-"Preferences"-"HotKeys". OBS Studio can mix, record and stream the final product over the Internet using a software-assisted configuration.

D. MIXING SOUND

As it has been mentioned before, for operational reasons, sound mixing must be performed on a computer other than the device with OBS Studio. Several approaches were tested to build an effective sound mixer. The first sound system on a computer with Linux Ubuntu 18 was able to send a sound mix to the computer with OBS Studio with a delay of approximately 7 s, which was unacceptably long.

To solve this long-latency issue, Linux 2017 was mounted with a special distribution, KX Studio, on the Lenovo Think Centre M58P computer with a 4-GB RAM. KX Studio is an operating system used in dedicated applications for sound production, whose kernel was specially manufactured to put priority on sound processes and audio communications. With this new setup, the delay in sound transmission to OBS Studio was reduced to 400 ms, approximately. After several tests, it improved up to 86 ms.

The overall sound setup for our experiments was a simple one, with some room for improvements. Two microphones were connected to the Eurorack UB1204FX-Pro soundboard listed in Table 3 via BobCat LAN sound extenders and computer network CAT6 cables, as shown in Fig. 6.

[19]Fade is a gradual visual transition between two video sources.

FIGURE 6. XLR sound extenders with LAN connectors.

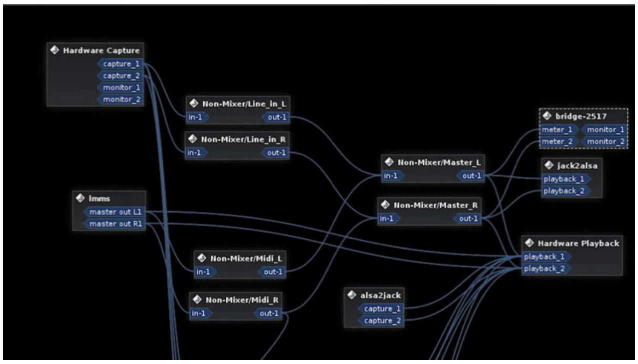

FIGURE 7. Claudia's sound connections interface.

The soundboard was connected directly to the sound card of the KX Studio computer, which is set up to be able to receive audio not only from the UB1204FX-Pro soundboard, but also from other applications playing audio on the same device, such as Linux Multimedia Studio (LMMS) [26].

KX Studio was built on a sound server called Jack, whose function is to minimize audio communication latencies. To implement these mixing functions, we have used the application Non-Mixer [27], which comes with this dedicated operating system by default. For the audio mixing to take place, the sound sources and audio output need to be linked to Non-Mixer. As shown in Fig. 7, this task is performed using the application Claudia [28].

Claudia allows users to link sound inputs and outputs freely with their mouse. For example, in Fig. 7, capture 1 and capture 2 receive microphone signals from the soundboard via two stereo channels. Capture 1 and capture 2 are sent to the Non-Mixer/Line inputs (L, R) to monitor stereo volume levels. Something of the sort can be done with the application LMMS, which has its own Non-Mixer/Midi volume control. The outputs from the line and midi Non-Mixer sources are then connected to the output of a master volume control called Non-Mixer/Master (R and L). The stereo output of the latter is linked to the application Jack [29] for external communication.[20]

Precisely, the sound produced by Non-Mixer must be sent to the computer with OBS Studio. To do this, a Linux command to forward the final audio mix has been created:

arecord -v -f cd | ssh user@1.0.1.2 'aplay -'

This command starts a remote session on the computer with OBS Studio and plays the mix automatically as desktop audio. The meaning of this command is the following: *record the sound mix with CD quality; display verbose status messages; open the mix simultaneously and remotely on the OBS Studio computer with IP address 1.0.1.2 as per user's credentials; and play it on the remote computer using the application Aplay*.

Aplay [30], an application on the OBS Studio computer, directly plays the sound from the audio mixer. OBS Studio identifies the input source as desktop audio, without further configuration being required (except for synchronization).

E. SYNCHRONIZATION OF SOURCES

Synchronization for television broadcasting over digital networks is a problem addressed by different historical developments [31]. It is related to the simultaneous reception of media content [32]. However, it must be mentioned that film synchronization challenges also existed in the analog or even the celluloid era of media production.

Synchronization between sound and moving pictures, for example, has been key since the beginning of the film industry, as pictures and sounds are always expected to be delivered to the audience in a simultaneous way. This is one of the reasons why clapperboards have always been useful, even nowadays [33]. The typical confirmation on shooting sessions *"Light, camera, action"* is traditionally followed by a click sound from a clapperboard.

When shooting for postproduction (editing), the click sound of the clapperboard allows video editors to synchronize pictures and sound, using recordings from different devices, for example. Synchronization can also occur between different camera sources. Clapperboards offer sound-only and visual-only synchronization too. When using different cameras, the exact moment when the two pieces of wood (or any other modern material) touch each other should be visually synchronized for all the cameras in that shooting. They should also be synchronized with the clapperboard click sound.

In live productions, this technique is difficult to use. Nevertheless, both visual and sound synchronization are a key issue that needs to be addressed. Although it must be noted that audiovisual synchronization involves large theoretical considerations in the digital era [34], these are out of the scope of our paper.

Fig. 8 shows the basic setup we have created to visually synchronize camera sources. The cameras are rotated to frame the two monitors on the desk. The monitor on the right shows a millisecond web-based counter [35], whereas the monitor on the left displays the output of the OBS Studio mixing.

By using OBS Studio on its recording mode and switching from camera to camera, the delay can be calculated on the basis of the counter information on both screens. Fig. 9, which

[20]Bridge-2517, in Fig. 7, is a graphic monitor used for control purposes only.

FIGURE 8. Visual synchronization setup.

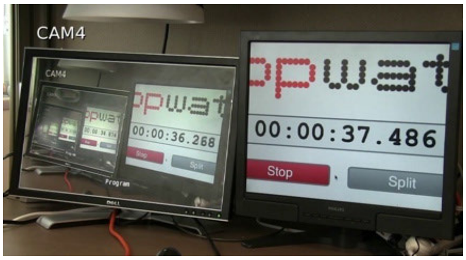

FIGURE 9. Example of output delay observation.

is a zoomed-in picture of Fig. 8, illustrates delay in CAM4, which seems to be the largest in our experiments. The delay identified in CAM4 can be simply calculated from Fig. 9 as follows: 00:00:37:486 - 00:00:36:268 = 1218 ms.

The technique used for synchronizing camera sources includes the activation of the OBS Studio Video Delay (Async) filter (by right-clicking the media source camera) and the addition of extra delays to match the longest delay identified (CAM4).

For sound, synchronization is performed based on the source with the longest delay (CAM4) by trying to match voices and lip movement. Using a video editing software, like Adobe Premiere Pro CC [36], the delay is identified by following routine filmmaking procedures in two steps (Fig. 10).

In Fig. 10, the track that reads "Colour_red" shows for how long sound had to be delayed: 1275 ms.[21] This audio delay was then inserted in OBS Studio by going to "Advanced Audio Properties", "Desktop Audio Sync offset (mix)". Table 7 summarizes the delays artificially inserted in OBS Studio to match the CAM4 source. We assumed that CAM6 (OpenLP) did not require synchronization.[22]

In terms of total video delay, Fig. 9 shows that CAM4 has a delay of 1 s and 218 ms, or 1218 ms, which has been caused by video images. The remaining cameras are synchronized based on these data.

[21] This is the equivalent in milliseconds that we have obtained using Adobe Premiere's "duration" feature. This information has been recovered by right-clicking the "Colour_red" block in Fig. 10..

[22] This assumption basically means that OpenLP is used mostly for titles or still pictures, and not for playing synchronized videos and sound. To play videos with audio, a buffer filter must be added to the CAM6 sound source on KX Studio.

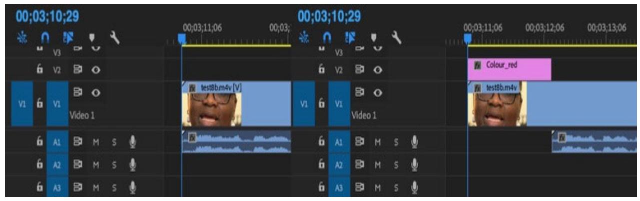

FIGURE 10. Determination of sound offset on video editor.

TABLE 7. Added buffer delay on OBS studio.

Source	Added delay
CAM1	230 ms
CAM2	57 ms
CAM3	232 ms
CAM4	0 ms
CAM5	173 ms
Sound	1275 ms

For audio, as it has been mentioned before, the sound delay in communications between KX Studio and OBS Studio was 86 ms, approximately, at its best. As shown in Table 7, a sound buffer of 1275 ms was added to the system, thus causing a total sound delay of 1361 ms (1275 ms + 86 ms). The sound delay was therefore 143 ms (1361 ms − 1218 ms) higher than the video delay.

As a result, the difference between the video delay and the audio delay was four video frames.[23] Such a difference might be due to accuracy issues when manually determining synchronization between voices and lip movement in a video editing software, for example. Therefore, it is correct to estimate a total system delay of 1.4 s, approximately.

F. MULTITRACK AND MULTICAMERA RECORDING

In terms of live streaming, the system, as presented at this stage of our research, could provide multicamera and multisound source mixes. Nevertheless, if media sources were to be remixed in postproduction, some additional functions would be required. For instance, this might be the case in music concerts, where rhythmic video mixes are expected to match the music and more elaborated editing techniques are thus required.

In these situations, video servers (PC router 1 and PC router 2) need to be slightly modified. On each Linux Debian virtual machines, the following command must be executed using a new Linux window:

socat udp4-recv:5004,bind=239.255.42.42,ip-add-membership=239.255.42.42:enp0sX, reuseaddr -> camX.ts

where X is the video source (1 to 6).

Also, KX Studio requires a dedicated application to handle multitrack sound recording: Ardour [37]. Ardour is

[23] Considering 30 fps (or 30 frames/1000 ms) using the previously described cameras, 143 ms would be equal to 143 ms x 30 frames / 1000 ms = 4.3 frames (approximately four frames).

a multitrack audio production software that records different sound sources and keeps them synchronized with its output.[24]

G. VOICE INTERCOMMUNICATION

Any live production including multiple cameras requires an intercommunication system, so that team members are able to work in a coordinated way. Most of these communications are oral, such as those that take place between a program director and camera operators.

The application Unity Intercom has been used to provide voice intercommunication between the members of the audiovisual production team. This application runs on mobile devices with Android or iOS.

In addition, the network has been expanded to cover Wi-Fi using the D-Link AC1750 device, which does not require a specific configuration. A voice server has also been downloaded [38] and installed on the Macbook Pro 13-in Retina computer.

This system, which is already available in the market, has helped us reduce the cost of technology in our system, so that regular mobile phones could be used instead of expensive dedicated intercom devices. An alternative application could have been programed as well.

H. TESTS

Several tests have been carried out on long live streaming and multicamera recording of events to assess the autonomy of the system. Due to extension limitations, it would be impossible at this time to detail all the variables considered and adjusted during the tests. But they had in consideration quantity of movements in cameras, audio source traffic and voice intercommunication.

Cameras have been switched automatically using keyboard actions during long periods of time. The keyboard actions were customized using the application Actiona [40]. Those tests showed that the system performed normally for around six hours at a maximum. After that relative long time for video production, some camera sources became still pictures, due to system resource limitations, and needed to be reloaded as a media source to show movement again.

V. CONCLUSION

Like any other engineering project, this research has encountered several challenges along the way. Some of them have been related to how to effectively provide the video mixer with the appropriate video sources without creating interference, what applications to use to mix sound and video, how to synchronize multimedia sources, how to integrate a voice intercommunication mechanism to the system, among others.

To solve all these problems, a complex network has been designed. Open-source applications have been configured, so that they emulate a live TV production mixer: OBS Studio for video and Linux KX Studio for sound. Finally, the network has been expanded to provide Wi-Fi access and use Unity Intercom as a voice intercommunication system.

The objective of this paper was to integrate different COTS and mainly open-source tools to create a low-budget system for live productions. Our results demonstrate the feasibility of such an integrated system.

As a general conclusion, it must be said that the video-audio production mixer via an IP network described here has fulfilled the designers' expectations. It provides *real-time* video and sound mixing with an acceptable delay of 1.4 s, approximately. This total delay complies with the condition for multicamera productions (recording or live streaming) presented in Table 2, according to which the delay should be of less than 3 s, based on professional production experiences, if there were no projections on a big screen near the stage nor a center of captured action (a situation that would require a system delay of less than 1 s based on QoE assumptions).

The 1.4-s delay can be considered reasonable, particularly as non-dedicated hardware has been used. In video applications, dedicated hardware plays a significant role, like in 3D real-time renderings [39] or even during image processing with the Ross Video mixer mentioned in Section II-B.

The system saturation tests detailed in Section IV-H, indicated that the system works properly for a maximum of six hours of continuous multicamera recording or live streaming. The system is thus convenient for a typical live production, which is assumed to last less than six hours.

The video-audio production mixer via an IP network ensures operation of up to five high-definition camera sources, a title generator, multimedia playback, music, microphone sound, among other components. This system has been implemented using any type of HDMI cameras, soundboards, microphones, and other available equipment. A few dedicated devices are required, including HDMI to IP transmitters, and IP sound extenders. It is a low-budget system, ideal for applications such as community television station productions or small-to-medium-sized budget live productions.

The utility of the system described here has been largely proved. It has been, adopted by Eidei Productions Corporation, a non-profit audiovisual production company from Canada [41].

When considering future research, one of the features that could be added to the system is a *tally*. A tally is a kind of light located at the front and at the back of a camera that warns camera operators and performers when that camera source has been selected for video mixing.[25] The functions of a tally must be supported by the video mixer. OBS Studio supports this feature via a series of plugins known as OBS Websockets [42]. Open-source applications that display the tally functions, like Pro Tally [43], could be included as well.

[24]This means that all the track recordings start at the same time, thus reducing postproduction (editing) work. This is not the case of the Socat command used for recording video, since those commands are executed separately.

[25]Tally lights are usually located close to the camera viewfinder, so that operators can easily see the signals. Sometimes, a small LED light is enough. The front camera tally light is usually more powerful and visible than the back camera tally light; hence, performers can identify when they are on or off.

If that were the case, special hardware would be necessary to create a light that could be activated via Websockets.

As it has been mentioned in Section II-A, an innovative line of research on the video-audio mixer via an IP network could be the use of AI in camera movements, sound mixing operations, and oral instruction recognition. As a result, a single director would be able to direct both the video and the sound mixing, like the BBC almost real-time mixer described in that section. Camera operations and sound mixing via AI would also increase the artistic quality of productions, while keeping budgets affordable to television community stations and the like.

Another potential future development for this system could be assessing the possibility of adopting teleproduction. Teleproduction means that the control area of a production is located in a central location, whereas the production area is situated in a remote location (out of town, a different region or country). As a result, only camera operators and some technicians are required to work at the shooting or transmission location, while the director and the video mixer can be located in a different central location. Some companies already offer communication services to support teleproduction [44]. The system described in this research would require some communication adaptations and latency considerations if teleproduction were adopted.

REFERENCES

[1] J. Owens, "The second screen," in *Television Production*. London, U.K.: Focal Press, 2015.
[2] A. Utterback, *Studio Television Production and Directing*. Waltham, MA, USA: Focal Press, 2015, pp. 1–5.
[3] Open Broadcaster Software. *OBS Studio Community*. Accessed: Apr. 6, 2019. [Online]. Available: https://obsproject.com
[4] KX Studio community. *KX Studio*. Accessed: Apr. 17, 2019. [Online]. Available: https://kx.studio
[5] L. Fay, L. Michael, D. Gomez-Barquero, N. Ammar, and M. W. Caldwell, "An overview of the ATSC 3.0 physical layer specification," *IEEE Trans. Broadcast.*, vol. 62, no. 1, pp. 159–171, Mar. 2016.
[6] C. Akamine, R. Martins, R. A. Vaz, A. Godoi, G. H. M. G. De Oliveira, R. S. Rabaca, U. M. R. E. Barros, R. Brancaccio, D. Ohata, J. R. Ribeiro, C. Diez, J. R. Soares, L. G. P. Alves, F. P. Del Nero, G. Ferraresso, L. Chaves, and P. Notargiacomo, "Broadcast-Broadband experience in ATSC 3.0: Proposal for gamification and complementary cameras," in *Proc. IEEE Broadcast Symp. (BTS)*, Arlington, TX, USA, Oct. 2018.
[7] BBC. (2014). *Nearly Live Production-BBC (R&D)*. Accessed: Aug. 27, 2019. [Online]. Available: https://www.bbc.co.uk/rd/projects/nearly-live-production
[8] R. Descartes, *Meditations On First Philosophy: The Philosophical Works of Descartes*. Cambridge, U.K.: Cambridge Univ. Press, 1911.
[9] D. Luzuriaga, "El concepto de tiempo en la comunicación del espacio virtual televisivo The concept of time in the communication of the television virtual space," Ph.D. dissertation, Fac. Educ. Social Sci., Universidad del Salvador, Buenos Aires, 2008. [Online]. Available: https://racimo.usal.edu.ar/4471/
[10] Techopedia. *What is Quality of Experience (QoE)?* Accessed: May 29, 2019. [Online]. Available: https://www.techopedia.com/definition/25802/quality-of-experience-qoe
[11] K.-T. Chen, C.-C. Wu, Y.-C. Chang, and C.-L. Lei, "A crowdsourceable QoE evaluation framework for multimedia content," in *Proc. 17th ACM Int. Conf. Multimedia-(MM)*, Vancouver, BC, Canada, 2009.
[12] S. Baraković and L. Skorin-Kapov, "Survey and challenges of QoE management issues in wireless networks," *J. Comput. Netw. Commun.*, vol. 2013, pp. 1–28, 2013.
[13] Ross Video. *Ross Cabonite Black Solo*. Accessed: Jun. 7, 2019. [Online]. Available: https://documentation.rossvideo.com/files/Brochures/Production%20Switchers/Carbonite%20Black%20Solo/Carbonite%20Black%20Solo%20Brochure.pdf.
[14] V. Kompella, J. Pasquale, and G. Polyzos, "Multicast routing for multimedia communication," *IEEE/ACM Trans. Netw.*, vol. 1, no. 3, pp. 286–292, Jun. 1993.
[15] C.-H. Lung, A. Matrawy, and I. Lambadaris, "A framework for MPLS path setup in unidirectional multicast shared trees," *Proc. SPIE, Soc. Photo-Opt. Instrum. Eng.*, vol. 559, pp. 32–39, Oct. 2004.
[16] A. Matrawy, W. Yi, I. Lambadaris, and C.-H. Lung, "MPLS-based multicast shared trees," in *Proc. 4th Annu. Commun. Netw. Services Res. Conf. (CNSR)*, Moncton, NB, Canada, May 2006.
[17] M. Rana, M. Kaykobad, and A. B. M. A. Al Islam, "A new approach for selecting aggregated multicast trees to reduce forwarding states," in *Proc. 5th Int'l Conf. Netw., Syst., Secur.*, Dec. 2018, pp. 1–9.
[18] Z. Lin, "Research on video traffic control technology based on SDN," in *Proc. Joint Int. Mech., Electron. Inf. Technol. Conf.*, Chongqing, China, 2015.
[19] Wireshark. *Wireshark. Go Deep*. Accessed: Apr. 6, 2019. [Online]. Available: https://www.wireshark.org/#learnWS
[20] VideoLAN. *Official Download of VLC Media Player*. Accessed: Aug. 28, 2019. [Online]. Available: https://www.videolan.org/vlc/
[21] T. Nolle, "Location vs. Address: Tunnels, VLANs, and VPNs," *Netw. Mag.*, vol. 14, no. 9, p. 120, Sep. 1999.
[22] D-Link. *D-Link DGS-1100-24 User Manual*. Accessed: Apr. 9, 2019. [Online]. Available: ftp://ftp.dlink.ca/PRODUCTS/DGS-1100-24/DGS-1100-24_REVA_MANUAL_1.04_EN.PDF
[23] D-Link. *D-Link DSR-250 User Manual*. Accessed: Apr. 9, 2019. [Online]. Available: ftp://ftp.dlink.ca/PRODUCTS/DSR-500AC/DSR-SERIES_USER_MANUAL_3.12.pdf
[24] Oracle. *Oracle VM VirtualBox*. Accessed: Apr. 16, 2019. [Online]. Available: https://www.virtualbox.org
[25] *Socat-Dest-Unreach*. Accessed: Apr. 17, 2019. [Online]. Available: http://www.dest-unreach.org/socat/
[26] LMMS. *Linux Multimedia Studio*. Accessed: Aug. 28, 2019. [Online]. Available: https://lmms.io
[27] J. M. Liles. *Non Mixer User Manual*. Accessed: Mar. 30, 2019. [Online]. Available: http://non.tuxfamily.org/mixer/doc/MANUAL.html#n:1.1
[28] *KXStudio: Applications: Claudia*. Accessed: Aug. 28, 2019. [Online]. Available: https://kx.studio/Applications:Claudia
[29] Jack Audio Connection Kit. *Jack Audio*. Accessed: Aug. 28, 2019. [Online]. Available: http://jackaudio.org
[30] Alsa Open Source. *Aplay-ALSA*. Accessed: Aug. 28, 2019. [Online]. Available: https://alsa.opensrc.org/Aplay
[31] P. H. Lawrence, B. W. Dunn, and M. A. Eshleman, "System and method for maintaining timing synchronization in a digital video network," U.S. Patent 6 208 666 B1, Mar. 27, 2001.
[32] Z. Huang, K. Nahrstedt, and R. Steinmetz, "Evolution of temporal multimedia synchronization principles," in *MediaSync: Handbook on Multimedia Synchronization*. Cham, Switzerland: Springer, 2018, pp. 33–71.
[33] T. Holman and A. Baum, "Syncing Footage," in *Sound for Digital Video*. London, U.K.: Focal Press, 2013.
[34] H. Melvin, J. Shannon, and K. Stanton, "Time, frequency and phase synchronisation for multimedia-basics, issues, developments and opportunities," in *MediaSync: Handbook Multimedia Synchronization*. Cham, Switzerland: Springer, 2018, pp. 105–146.
[35] Stopwatch. *Stopwatch on line counter*. Accessed: Apr. 6, 2019. [Online]. Available: https://www.estopwatch.net
[36] Adobe. *Adobe Premiere Pro*. Accessed: Aug. 28, 2019. [Online]. Available: https://www.adobe.com/ca/products/premiere.html
[37] Ardour. *The Ardour Manual*. Accessed: Apr. 17, 2019. [Online]. Available: http://manual.ardour.org/toc/
[38] *Download Unity Server Intercom–Unity Intercom*. Accessed: Apr. 4. 2019. [Online]. Available: http://www.unityintercom.com/download-landing-page
[39] H. Li, P. An, and Z. Zhang, "Full high-definition real-time depth estimation for three-dimensional video system," *Opt. Eng.*, vol. 53, no. 9, Sep. 2014, Art. no. 093108.
[40] J. Mercier-Ganady. (2009). *Actiona-Jmgr's Website*. Accessed: Mar. 28, 2019. [Online]. Available: https://www.jmgr.net
[41] Eidei Productions Corporation. (Mar. 2011). *Non-For Profit Production Company Canada Incorporated 779655-2*. Accessed: Aug. 28, 2019 [Online]. Available: http://www.eidei.tv

[42] S. Palakis. *Github-OBS Websockets API for OBS Studio*. Accessed: Apr. 5, 2019. [Online]. Available: https://github.com/Palakis/obs-websocket
[43] J. Adams. *Github-Pro Tally: On Screen Tally Software*. Accessed: Apr. 4, 2019. [Online]. Available: https://github.com/josephdadams/ProTally
[44] Nevion. *Nevion Virtuoso-Software Defined Media Node*. Accessed: Apr. 6, 2019. [Online]. Available: https://nevion.com/virtuoso/

DARIO LUZURIAGA received the B.A. degree in television production from the University of Belgrano, the Ph.D. degree in social communication sciences specialized in T.V. journalism from the University del Salvador, Buenos Aires, in 2008, and the B.Eng. degree specialized in communications engineering from the Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada, in 2019, He is a former Assistant Professor of production and direction of television and radio shows from the University of Belgrano, from 2001 to 2008. He extended his education to philosophy with the University of Buenos Aires. He was a Media Producer for several radio and television stations in Argentina, from 2001 to 2008, including America 24 TV and Rivadavia Radio, contributing to the Martin Fierro Award. He was recruited by the Government of Quebec and moved to Canada. He was a Filmmaker and a Postproduction Chief of select productions and select audio-visual, from 2009 to 2010. In 2011, he founded Eidei Productions Corporation, where he is currently the CEO.

CHUNG-HORNG LUNG received the B.S. degree in computer science and engineering from Chung Yuan Christian University, Taiwan, and the M.S. and Ph.D. degrees in computer science and engineering from Arizona State University. He was with Nortel Networks, from 1995 to 2001. In September 2001, he joined the Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada, where he is currently a Professor. His research interests include computer networks, software engineering, and cloud computing.

MARGARET FUNMILAYO received the B.Sc. degree in physics with electronics concentration from Ajayi Crowther University, Nigeria, in 2014, and the B.Eng. degree in communications engineering from Carleton University, Canada, in 2019.