IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 16 May 2023, accepted 3 July 2023, date of publication 12 July 2023, date of current version 4 August 2023.

*Digital Object Identifier 10.1109/ACCESS.2023.3294561*

RESEARCH ARTICLE

# Study on Eliminating Delay and Noise in On-Site Audio Center of Anchor Technology

**SEUNG-WON LEE[1], SOON-CHUL KWON[2], AND SEUNG-HYUN LEE[3]**
[1]Department of Immersive Content Convergence, Kwangwoon University, Seoul 01897, Republic of Korea
[2]Graduate School of Smart Convergence, Kwangwoon University, Seoul 01897, Republic of Korea
[3]Ingenium College Liberal Arts, Kwangwoon University, Seoul 01897, Republic of Korea

Corresponding author: Seung-Hyun Lee (shlee@kw.ac.kr)

This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant funded by the Korean Government (MSIT) (Development of Holographic Stereogram Printing Technology Based on Multi-View Imaging) under Grant 2020-0-00922; in part by the IITP Grant funded by the Korea Government (MSIT) (Research and Development of Realistic Content Device Technology) under Grant 2020-0-01846; in part by the Ministry of Culture, Sports and Tourism; in part by the Korea Creative Content Agency under Project R2021040083; and in part by the Excellent Researcher Support Project of Kwangwoon University, in 2022.

**ABSTRACT** Audio over IP (AoIP)-based audio transmission technology, which has recently been introduced into the audio market, has led to the development of conventional audio technology. Among the AoIP-based technologies, the audio networks and control hierarchy over peer-to-peer (Anchor) system technology converts audio signals into digital signals and enables real-time transmission and reception without audio loss to a desired location through Internet protocol (IP). In addition, on the receiving side, it allows selecting and mixing multiple transmission signals (audio sources) and controlling transmission and reception signals. The Anchor system replaces the conventional audio mixer and creates a novel paradigm in audio system configuration as it can control transmission and reception signals at a desired location with broadband. However, a delay occurs intermittently or noise is issued at the receiving side of On-site audio center (OAC), a device that transmits and receives audio data. In this study, causes for delay and noise have been analyzed and eliminated through data packet processing in a receiving OAC according to Anchor technology. As a result of the cause analysis, it has been confirmed that there is a data loss due to delay in audio data processing in an audio codec buffer of the receiving OAC. This problem has been solved by eliminating the delay and noise through a controlling algorithm for handling an audio data play speed. As a result of the study, it has been confirmed that high-quality clear audio is played for a long time without delay and noise. It is expected that Anchor technology will be established as a standard for industrial audio system facilities using AoIP technology in the future.

**INDEX TERMS** Anchor, AoIP, audio mixer, Dante, OAC.

## I. INTRODUCTION

Audio over IP (AoIP) technology used in industrial audio systems has been released for various products in the audio market and is facing an inflection point based on novel changes and developments. Dante technology, which is leading the AoIP market, can convert audio signals into digital and transmit and receive Internet protocol (IP)-based transmission; it is used in systems that can transmit long distances or transmit some sections in multiple channels and real-time [1], [2]. It can reproduce high-quality audio without loss by replacing the audio cables; however, it does not change the audio system configuration. In contrast, audio networks and control hierarchy over peer-to-peer (Anchor) technology enable a novel audio system configuration different from the conventional one. Anchor technology is a novel technology recently released into the market that transmits and receives audio signals over broadband based on IP.

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

**TABLE 1. Comparison of audio system based on AoIP.**

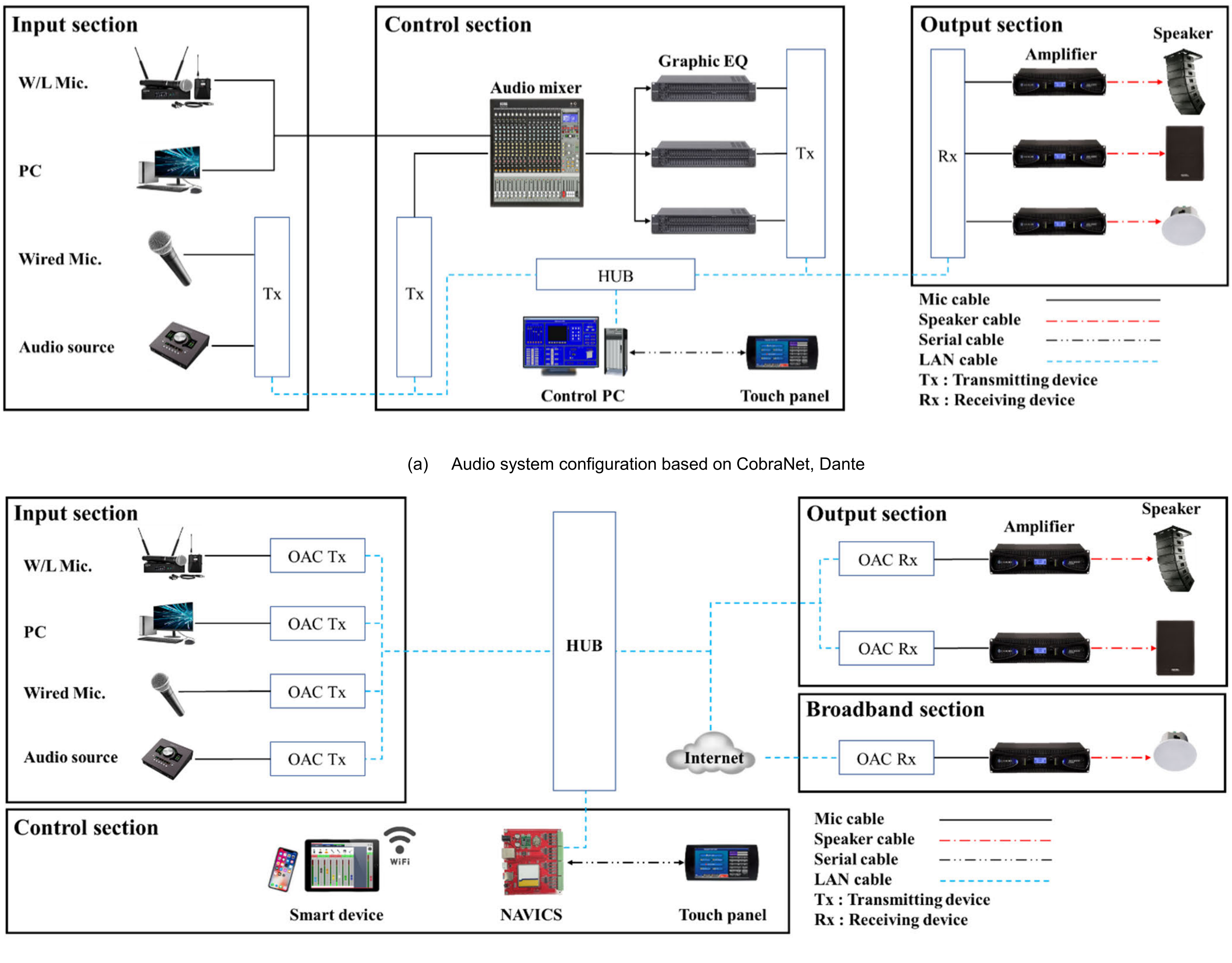
| No | Technology | Year | Transport | Transmission Scheme | Control Communications | Network Capacity | Latency | Sampling Rate | Mixing |
|---|---|---|---|---|---|---|---|---|---|
| 1 | CobraNet | 1996 | Ethernet data link layer | Isochronous | Ethernet, SNMP, MIDI | More than 32 | 1 1⁄3, 2 2⁄3 and 5 1⁄3 ms | 96 kHz | Network = No Independence = No |
| 2 | Dante | 2006 | Any IP medium | Isochronous | IP, Bonjour | More than 32 | 84 µs or greater[h] | 48 kHz | Network = Yes Independence = No |
| 3 | AVB | 2011 | Enhanced Ethernet | Isochronous | IEEE 1722.1 | More than 32 | 2 ms | 192 kHz | Network = Yes Independence = No |
| 4 | Arria Live | 2013 | Ethernet | Isochronous | Ethernet | 32 CH(mono) | 2.3 ms | 48 kHz | Network = No Independence = Yes |
| 5 | Anchor | 2017 | Ethernet (DIX Ethernet) | asynchronous | IP, UDP | 16 CH(stereo) | 10.7 ms | 48 kHz | Network = Yes Independence = Yes |

In addition, an audio system can be configured without an audio mixer using a dedicated transmitting and receiving device [3]. Since Anchor technology can replace conventional audio mixer-centered systems, it will lead the audio industry by enabling a novel audio system configuration. However, the Anchor technology often faces problems of signal delays and noise on the receiving side when transmitting and receiving audio signals. This study proposes a method using Anchor technology to transmit and receive audio signals over broadband based on IP and eliminate the delay and noise at the Anchor's On-site Audio Center (OAC). Section II describes the Anchor technology, and Section III describes eliminating delay and noise from the Anchor's OAC. Section IV presents the experimental results. Section V discusses the conclusions and future applications based on Anchor technology.

## II. ANCHOR TECHNOLOGY

Advances in AoIP technology began with the technologies for the long-distance and high-quality transmission of audio signals. Recently, this has enabled a novel configuration of audio systems to reduce equipment costs and enable efficient operation [1], [4]. In particular, the Anchor technology, launched in 2017, enables local system configurations to become broadband configurations, representing a novel paradigm in the audio industry. Table 1 compares the characteristics of several AoIP technologies in the market. CobraNet is one of the technologies that transmit audio signals of up to 64 channels in real time through a standard Ethernet network. Because CobraNet uses an Ethernet network, it can only be used on an internal network and communicates with network synchronization through a bundle, which is a unique package [2], [5]. Dante is a lossless compression multichannel audio transmission standard that transmits audio based on IP. Dante can transmit 48 $\times$ 48 channels of audio signals bi-directionally at 24-bit, 48 kHz over a single-link 100 Mbps Ethernet [3], [6]. Audio/Video bridging (AVB) is a technology that can transmit high-quality audio/video and control signals simultaneously and can be applied to industrial audio systems and public address systems [5], [7]. AVB transmits and receives data in real time by compressing audio/video signals and has become an international standard at the Institute of Electrical and Electronics Engineers (IEEE) [8], [9]. Arria Live transmits and receives audio signals over Ethernet through dedicated input/output modules and smart devices without an audio mixer. The audio signal input/output of Arria Live supports a mono 32 ch. An Anchor is a technology that transmits audio signals using IP on a network basis. It can mix and control transmission signals on each receiving device and configure an audio system without an audio mixer. Moreover, it is possible to configure a broadband system by transmitting and receiving IP-based audio signals [3], [10]. Arria Live and Anchor enable independent audio mixing on the receiving side of audio signals, enabling a novel audio system configuration that differs from conventional audio system configuration methods [11], [12].

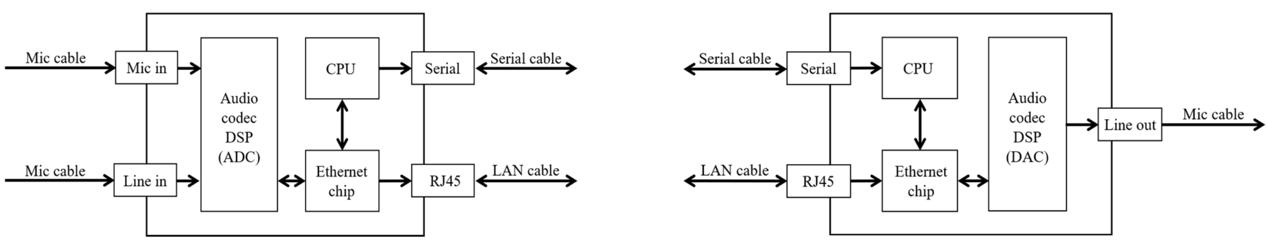
Fig. 1 compares typical audio system configurations based on AoIP technology. Fig. 1(a) and 1(b) show the CobraNet Dante-based and Anchor-based audio system configurations, respectively. In Fig. 1(a), the audio signals are transmitted using the AoIP technology for certain sections (from the audio input to the control sections and from the control to the audio output sections). However, the CobraNet and Dante technologies are existing audio system configurations centered on audio mixers that do not change the conventional system configuration [13], [14]. As shown in Fig. 1(a) and 1(b), the audio signals transmitted and received through an input/output device are the same. However, Anchor technology replaces the hardware equipment for control, such as audio mixers and graphic equalizer with software and allows controlling and mixing the audio input and output signals through the OAC [3]. Additionally, it is possible to build a broadband audio system through a network by changing the configuration of a conventional audio system [15]. Anchor technology consists of an OAC that




(a) Audio system configuration based on CobraNet, Dante

(b) Audio system configuration based on Anchor

**FIGURE 1.** **Comparison of audio system configuration diagram.**




(a) Transmitting OAC

(b) Receiving OAC

**FIGURE 2.** **OAC block diagram.**

transmits and receives audio signals and a network-based audio/video integrated control system (NAVICS) that can control the OAC [16].

Fig. 2 shows the transmitting/receiving signal processing diagram of the OAC. The transmitting OAC receives a microphone or line signal as the input. Fig. 2(a) shows the signal output of the audio source device, which is converted from analog to digital in the audio codec. The CPU determines the audio format (PCM), controls the audio volume and mutes, and records the IP and port information of the received OAC in its EEPROM. The Ethernet chip converted the audio data from the CPU into Ethernet II (DIX Ethernet) frame

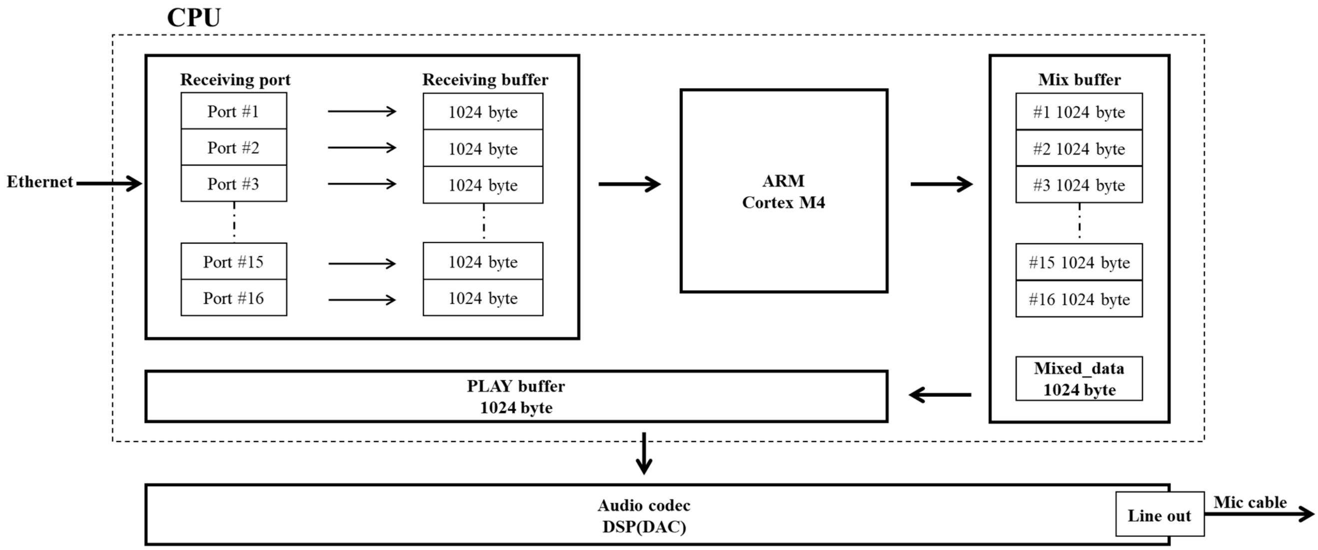



**FIGURE 3.** The receiving OAC's data processing diagram (CPU).

packets and transmitted them at 100 Mbps. Fig. 2(b) shows signal processing in the reverse order of (a). The transmitting OAC sends audio data by specifying the destination IP as a multicast IP, and the receiving OAC receives audio data by connecting to the multicast IP. The receiving OAC mixes audio data when there are two or more connected multicast IPs [17], [18], [19].

Fig. 3 shows a conceptual diagram of the CPU signal processing of the receiving OAC. The received audio data were stored in the buffer of the corresponding port by calling the callback function of the specified port. Subsequently, ARM Cortex M4 calls the mix function, which controls the audio signal-level data of the corresponding buffer. The mix function refers to the starting address of the stored receiver buffer and performs mixing. ARM Cortex M4 calls the play function, which passes mixed data to the audio codec [20], [21].

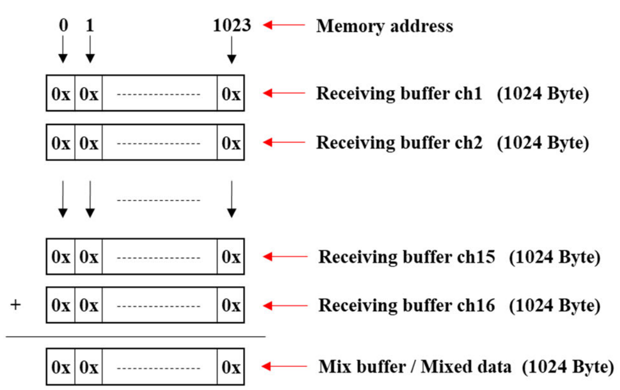



**FIGURE 4.** Mixing concept diagram in the receiving OAC.

Fig. 4 shows a conceptual diagram of the mixed audio data stored in the receiving buffer of the receiving OAC. The hexadecimal data stored in the address of each receiving buffer are converted into an audible sine-wave audio signal through decoding. Next, audio-level control was achieved by adjusting the ratio of the hexadecimal data. Finally, mixed final audio data were created by adding the hexadecimal data stored in each address of the receiving buffer. The audio-source-level control and mixing process can be expressed using a programming language as follows:

For (i=0, i<1024, i=i+1)

$$\text{Mixbuffer}\,[\text{i}] = \sum\nolimits_{ch=1}^{16} \text{Receivingbuffer}\,[\text{ch}]\,[\text{i}] * volume[ch]. \tag{1}$$

The volume[ch] in the program line is the ratio for each channel that controls the audio source signal level (sine-wave amplitude) [22]. The data processing of the program line controls the size of the hexadecimal data stored in 16 receiving buffers, adds the data of the same address value from addresses 0 to 1023 of the 16 receiving buffers, and stores them in a mixed buffer. The 16 receiving buffers are audio sources from 16 IPs [3]. Through the aforementioned management process, the OAC transmits, receives, and mixes the audio data. However, when the receiving OAC receives and plays the data of the transmitting OAC, an echo occurs after 1 h. After approximately 4 h, a delayed phenomenon occurs when the two OACs are played simultaneously. In addition, irregular fine noise (data loss) is observed. Section III describes the steps taken to eliminate various problems (delay and noise) that occur while the OAC is transmitting and receiving.

## III. ELIMINATING DELAY AND NOSIE IN OAC

An OAC is a device that transmits and receives audio data, and the receiving OAC mixes and controls audio data from different IPs. The received OAC causes delay and noise in the processing of audio data. This section analyzes the data processing of the receiving OAC and eliminates the causes of delay and noise. The receiving OAC sends audio data from the mix buffer to the play buffer and from the play buffer to the audio codec. To analyze the state in which the audio data sent from the play buffer are processed in the audio codec, the sample rate fine-tuning and sample counter provided by the audio codec manufacturer are checked. In addition, a patch was added to analyze and adjust the processing status of the audio data to eliminate delays and noise. Sample rate fine-tuning was used to fine-tune the playback speed of the audio signals, and the sample counter was the accumulated frame count of 128 bytes played from the start to the present in the audio codec.

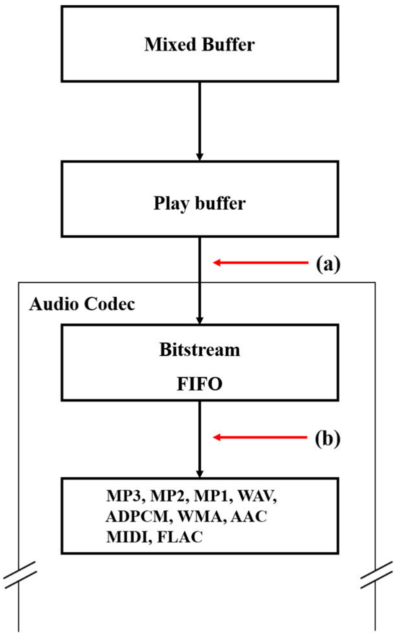



**FIGURE 5.** **Flow chart of the sample counter for the audio data.**

Fig. 5 shows a flowchart of the sample counter for the audio data of the receiving OAC. The sample counter analyzes the frames processed at points (a) and (b) in Fig. 5 to check the audio data processing and delay. Fig. 5(a) shows the point where frames are transmitted to the codec in the play buffer, and Fig. 5(b) shows the point where they are processed and outputted in the bitstream of the codec. In Fig. 5(a) and Fig. 5(b), the number of processed frames is analyzed, and the input/output frames of the bitstream are synchronized such that the frames can be processed in a constant state.

**Algorithm 1** Audio Data Frame Counter Algorithm

```
If (frame counter synchronous flag == 0)
        Currently Received Frame Counter = Current Playing counter;
        frame counter synchronous flag = 1;

Else
        Currently Received Frame Counter + = 1;
```

Algorithm 1 presents the algorithm for the audio data frame counter in pseudocode. If the frame counter synchronous flag is zero (the codec play counter and play buffer transfer counter have not yet been synchronized), the current playing counter from the codec is stored in the currently received frame counter variable. After saving, the frame counter synchronous flag is set to one such that synchronization is not performed thereafter. If the frame-counter synchronous flag is not zero, 128 bytes (one frame) are increased to the currently received frame counter and stored. The OAC compares the frame number sent from the play buffer to the codec with the play number from the codec. If there is a difference below a specific value, it slows down the play speed to increase the data in the buffer. In addition, if the difference exceeds a specific value, the play speed is increased to empty the buffer.

After analyzing the process of transmitting and playing audio data to the codec, we found that the following cases occurred: The OAC codec uses a 12.288 MHz crystal and operates the clock by increasing it quadruple internally. The crystal of each OAC was subject to clock errors owing to the tolerance of the components used and the ambient temperature. The transmitting OAC encodes 512 bytes approximately every 5 ms. When two receiving OACs are used simultaneously, the result is as follows:

-. The first receiving OAC decodes 512 bytes every 5.1 ms.
-. The second receiving OAC decoded 512 bytes every 4.9 ms, and the first receiving OAC had a processing error of 0.1 ms per block.
-. The data sent from the play buffer to the codec is 0.1 ms faster than the data processed by the codec.

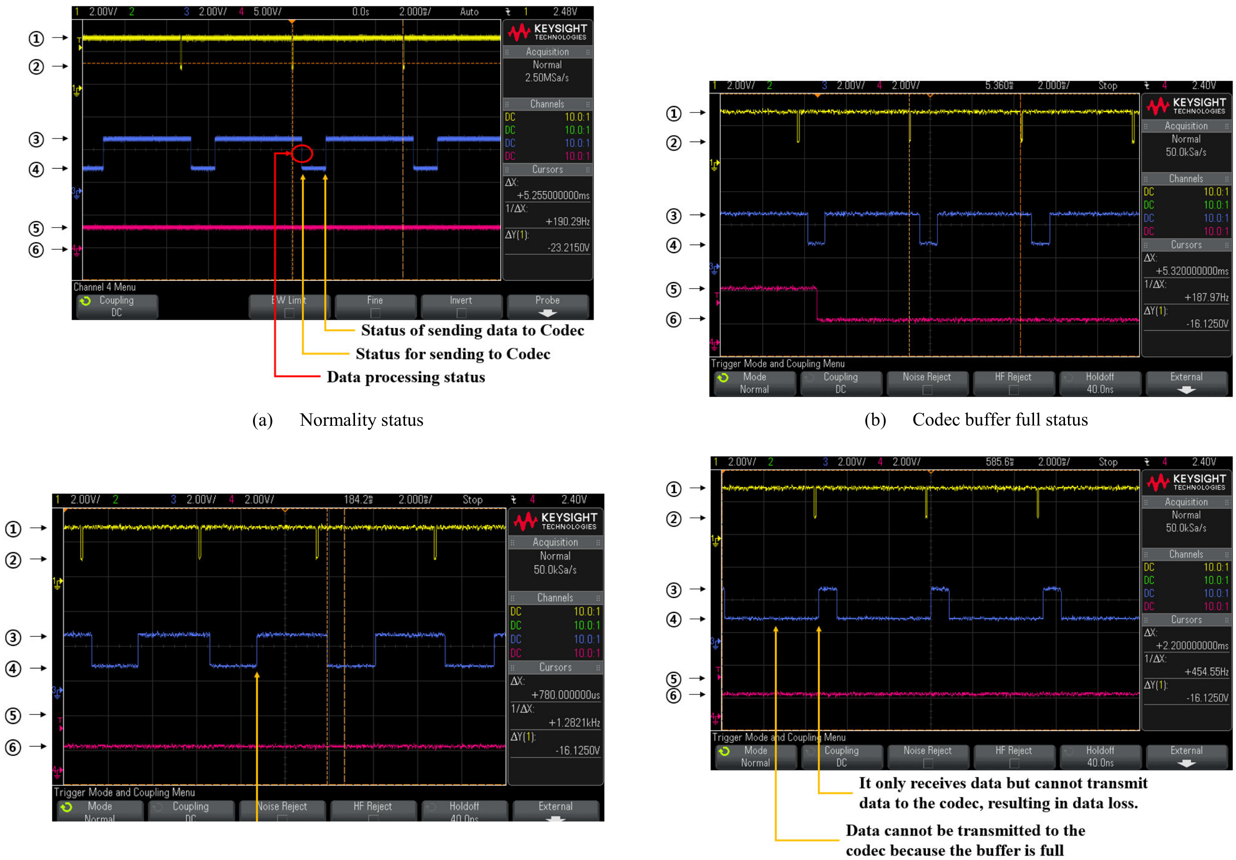
**Algorithm 2** Controlling Algorithm for Handling Audio Data Play Speed

```
Variable A = Cumulative number of frames sent to the codec
from play buffer
Variable B = Number of accumulated frames currently played
 by the codec

If [(Variable A) – (Variable B)] > 2 frames
    If (the current play speed mode is not fast)
        playspeed = 200;
        AdjustRate (ppm_playspeed);
    Else // The current play speed mode is fast
        Pass

 Else if [(Variable A) – (Variable B)] < 1 frame
    If (the current play speed mode is not slow)
        playspeed = –200;
        AdjustRate(ppm_playspeed);
    Else // The current play speed mode is slow
        Pass

 Else (When the number of frames is between 129 and 255)
   If (the current play speed mode is not normal)
    playspeed = 00;
    AdjustRate(ppm_playspeed);
   Else // The current play speed mode is normal
    Pass
```




(a) Normality status

(b) Codec buffer full status

(c) Data process delay status because the codec buffer is full

(d) Unable to process data status because the codec buffer is full

① **Transmission standby status through Ethernet of transmitting OAC**

② **Status of transmitting over Ethernet**

③ **Audio data processing standby state of receiving OAC**

④ **Data transmission status of Codec**

⑤ **Normal processing status of Codec Buffer**

⑥ **Codec buffer full**

**FIGURE 6. Bitstream audio data input/output analysis.**

-. If 50 blocks are sent from the play buffer to the codec, one block causes a delay (5 ms $\times$ 50 = 250 ms).
-. The codec has an Adaptive Differential Pulse Code Modulation (ADPCM) buffer that can store four blocks; after 250 ms $\times$ 4 = 1 s, the buffer will be full (a delay can be felt when the buffer is full).
-. The second receiving OAC generated a processing error of 0.1 ms per block.
-. The data sent from the play buffer to the codec is 0.1 ms slower than the data processed by the codec.
-. Transmitting 50 blocks from the play buffer to the codec
resulted in a shortage of one block (5 ms (1 block) $\times$ 50 = 250 ms).

To solve this problem, the audio data delay was solved using the following algorithm. Simultaneously, the noise problem was solved. Algorithm 2 shows the algorithm for handling the audio data play speed. The audio data rate processing was performed in three ways. In Algorithm 2, Variable A is the cumulative number of frames sent to the codec from the play buffer, and Variable B is the cumulative number of frames output from the codec bit stream. The first method is to increase the play speed by setting the speed to 200 when Variable A minus Variable B is greater than the number of frames normally processed and if the current play speed is not high. The second method is to slow down the play speed by setting the speed to −200 when Variable A minus Variable B is less than the number of frames normally processed and if the current play speed is not slow. The third method shows what should be processed when the frame processing is normal. The play speed is four parts per million (ppm) with a value of one, and the adjustment range is −187000 to 511999 based on a 24 kHz sample rate.

Delay and noise that occur in the receiving OAC can be eliminated through the controlling algorithm for handling the audio data play speed of Algorithm 2.

## IV. EXPERIMENTAL RESULT

The codec bitstream of the receiving OAC was analyzed through experiments on delay and noise processing using the speed control of audio data. The speed control analysis of audio data generally involves debugging through a time function or printF function; however, if analyzed in this way, delays occur due to function processing. Therefore, in this study, the audio data were analyzed through the high/low signals of the general-purpose input/output (GPIO) shown through the oscilloscope. The OAC's CPU operates at 168 MHz at the maximum clock speed; therefore, audio data analysis without delay is possible through the GPIO.

Fig. 6 shows analysis of audio data input/output of the bitstream in the receiving OAC. In each figure, ①, ②, ③, ④, ⑤, and ⑥ show a processing status of audio data. Fig. 6(a) shows a case where the status of audio data input/output of the bitstream of the codec in the receiving OAC is normal. ① shows a standby status for transmitting the audio data of transmitting OAC to Ethernet, ② shows a transmission status to Ethernet. ③ shows a standby status for audio data processing of the receiving OAC, and ④ shows a status of transmitting data from the play buffer to the codec. ⑤ shows a status in which the codec buffer is not full, and when the codec buffer becomes full, a status of ⑥ is brought. Fig. 6(b) shows a case where the status of ⑤ is changed to the status of ⑥ when the codec buffer becomes full. ⑥ in Fig. 6(c) indicates that the play buffer is processing data that is slightly delayed in being sent to the codec. Fig. 6(d) shows a status in which data of the play buffer cannot be received because the codec buffer is full, and shows a status in which the play buffer does not transmit data to the codec and receives next audio data. It has been confirmed that delay and noise occur due to audio data loss at this moment. In this study, delay and noise that occur due to the factors shown in Fig. 6(d) have been eliminated through the algorithm of Fig. 5 in Section III.

## V. CONCLUSION

This study aimed to eliminate the delay and noise caused by the receiving OAC of Anchor technology during transmitting and receiving audio data. This study analyzes the flow of audio data and describes eliminating delays and noise by controlling the data processing speed. This research enables the reproduction of a high-quality audio by eliminating noise and double audio generation factors. However, in order for Anchor technology to lead the market, there are challenges to be solved. If Anchor's transmitting OAC sends data to a different band, it is necessary to simplify the complicated routing process and shorten the audio data processing time (10.7 ms). The problems are expected to be solved with the development of network transmission technology and the improvement of CPU performance. Then, Anchor technology will lead AoIP-based technology. We expect Anchor technology to become the standard for AoIP-based audio system configurations in the future.

## REFERENCES


[1] C. Kang, S. Lee, K. Jo, and S. Lee, "Audio signal processing and system design for improved intelligibility in conference room," *J. Inst. Internet Broadcast. Commun.*, vol. 17, no. 2, pp. 225–232, Apr. 2017.

[2] M. S. Kang, S. W. Lee, and Y. S. Park, "Implementation of local distribution audio system based on AoIP," *J. Korea Inst. Inf. Commun. Eng.*, vol. 12, no. 12, pp. 2165–2170, 2008.

[3] J. Lee, J. Park, J. Lee, S. Kwon, and S. Lee, "Implementation of independent audio mixing control using AoIP," *Appl. Acoust.*, vol. 163, Jun. 2020, Art. no. 107235.

[4] M. S. Kang, K. Y. Sung, and Y. S. Park, "Design and implementation of audio transmission system based on AoIP," *J. Korea Inst. Inf. Commun. Eng.*, vol. 12, no. 8, pp. 1415–1419, 2008.

[5] J. W. Wee, Y. S. Park, K. W. Park, B. C. Song, and W. K. Jeun, "Audio/Video bridging technology for real-time AV transmission," *J. Korean Inst. Commun. Inf. Sci.*, vol. 30, no. 6, pp. 69–76, 2013.

[6] G. Kuzmics and M. Ali, "Audio networking in the music industry," *Ann. Emerg. Technol. Comput.*, vol. 2, no. 1, pp. 22–48, Jan. 2018.

[7] K. W. Park, D. H. Kim, K. W. Kwon, and B. C. Song, "IEEE 802.1AS performance test based on AVB (Audio/Video bridging)," in *Proc. Symp. Korean Inst. Commun. Inf. Sci.*, Feb. 2015, pp. 1342–1343.

[8] J. Wee, K. Park, K. Kwon, B. Song, and M. Kang, "Design and implementation of an embedded audio video bridging platform for multichannel multimedia transmission," *J. Internet Comput. Services*, vol. 16, no. 2, pp. 1–6, Apr. 2015.

[9] D. Maxim and Y.-Q. Song, "Delay analysis of AVB traffic in time-sensitive networks (TSN)," in *Proc. 25th Int. Conf. Real-Time Netw. Syst.*, Oct. 2017, pp. 18–27.

[10] J. H. Lee, H. J. Jeon, P. H. Choi, S. C. Kwon, and S. H. Lee, "Implementation of a novel audio network protocol," *Arch. Acoust.*, vol. 43, no. 4, pp. 637–645, 2018.

[11] E. S. Myakotnykh and U. P. Svensson, "Towards a computational quality model for IP-based audio," in *Proc. Int. Workshop Quality Multimedia Exper.*, Jul. 2009, pp. 110–115.

[12] T. Nunome, S. Tasaka, and K. Nakaoka, "Application-level and user-level QoS assessment of audio-video IP transmission over cross-layer designed wireless ad hoc networks," *IEICE Trans. Commun.*, vol. E91B, no. 10, pp. 3205–3215, 2008.

[13] N. Pekez, A. Popovic, and J. Kovacevic, "Ethernet TCP/IP-based audio interface for DSP system verification," *IEEE Consum. Electron. Mag.*, vol. 10, no. 1, pp. 45–50, Jan. 2021.

[14] J. He and W. H. Zhang, "Digital audio broadcast over IP: Problems and solutions," *Wireless Commun. Sensor Netw.*, pp. 821–828, 2016.

[15] J. H. Lee, S. C. Kwon, and S. H. Lee, "Implementation of on-site audio center based on AoIP," *Int. J. Adv. Smart Converg.*, vol. 6, no. 2, pp. 51–58, 2017.

[16] S. W. Lee, S. C. Kwon, and S. H. Lee, "A study on the audio/video integrated control system based on network," *Int. J. Adv. Smart Converg.*, vol. 11, no. 4, pp. 1–9, 2022.

[17] J. H. Lee, H. A. Carl, S. C. Kwon, and S. H. Lee, "System integration for the operation of unmanned audio center based on AoIP," *Int. J. Adv. Smart Converg.*, vol. 6, no. 2, pp. 1–8, 2017.

[18] M.-Y. Choi, S.-N. Lee, M.-J. Lee, Y.-H. Lee, H.-J. Park, and B.-S. Kong, "Multi-channel audio CODEC with channel interference suppression," *J. Semicond. Technol. Sci.*, vol. 15, no. 6, pp. 608–614, Dec. 2015.

[19] C.-M. Huang, T.-H. Hsu, and Y.-W. Lin, "REDUP: A packet loss recovery scheme for real-time audio streaming over wireless IP networks," *J. Syst. Softw.*, vol. 79, no. 1, pp. 29–42, Jan. 2006.
[20] S.-I. Kim, "Implementation of ARM based embedded system for muscular sense into both color and sound conversion," *J. Korea Contents Assoc.*, vol. 16, no. 8, pp. 427–434, Aug. 2016.
[21] D.-K. Kim and D.-Y. Yoon, "The design of low-cost vector-controlled inverter for induction motor using ARM cortex-M4 microcontroller," *J. Korea Academia-Ind. Cooperation Soc.*, vol. 14, no. 2, pp. 816–821, Feb. 2013.
[22] S. H. Ryu and H. G. Kim, "Audio mixer algorithm for enhancing speech quality of multi-party audio telephony," *J. Acoust. Soc. Korea*, vol. 32, no. 6, pp. 541–547, 2013.

**SEUNG-WON LEE** received the B.S. degree in media information engineering from Kangnam University, South Korea, in 2020, and the M.S. degree in immersive contents technology from Kwangwoon University, South Korea, in 2022, where he is currently pursuing the Ph.D. degree in immersive contents convergence. His research interests include immersive audio/video system design, acoustic engineering, and real-time integrated control systems.

**SOON-CHUL KWON** received the B.S. degree in industrial engineering from Hanyang University, in 2002, and the M.S. and Ph.D. degrees in information display from Kwangwoon University, in 2008 and 2012, respectively. He is currently an Associate Professor with Kwangwoon University. He has published over 40 journal articles in the related fields. His research interests include intelligent imaging processing, 3D graphics (virtual/augmented/mixed reality), hologram display, and human–computer interface. He serves as an Auditor for the ICT-Advanced Engineering Society.

**SEUNG-HYUN LEE** received the M.S. and Ph.D. degrees in holography and optical information processing from Kwangwoon University, Seoul, South Korea, in 1986 and 1992, respectively. He is currently a Professor with Kwangwoon University. His research interests include hologram printing technology, immersive content creation/editing/display technology, digital holographic terminal technology, VR/AR/MR/XR component implementation technology, HMD (head-mounted display) human factors, volumetric, photogrammetry content acquisition, intelligent image processing for surveillance and reconnaissance, and holographic display technology. He has published more than 100 journal articles in these fields. He is a Board Member of the International Hologram Manufacturers Association (IHMA). He serves as the Conference Chair for the SPIE. He is the Korean Representative of the International Stereoscopic Union and a Standardization Expert in ISO/TC172/SC9/WG7.