# Network Music Performance Beyond 4G

Konstantinos Tsioutas,* Yannis Thomas,* Fotios Bistas,* Ioannis Barous,*
George Xylomenos,* George C. Polyzos*†
* Mobile Multimedia Laboratory, Department of Informatics,
School of Information Sciences and Technology, Athens University of Economics and Business, Greece
† School of Data Science, The Chinese University of Hong Kong, Shenzhen, 518172 Guangdong, China

***Abstract*—Network Music Performance (NMP), where musicians perform music together over the Internet, requires ultra-low delays. Achieving a sense of presence in NMP requires both aural and visual communication, preferably coupled with immersive telepresence technologies, such as holographic communication. Existing 4G networks offer neither the ultra-low latencies needed for audio and video, nor the ample bandwidth required for video and (especially) holographic communication. In the Telepresence-Enhanced Network Music Performance (TENeMP) project we are investigating and developing solutions for immersive NMP, to be tested in the 5G testbeds of the SPIRIT project. We provide an overview of the experimental testbeds of TENeMP as well as the software tools developed by the project. We then present baseline video and audio latency measurements over 4G and 5G-NSA networks, which demonstrate NMP's need for the capabilities of 5G-SA and beyond networks.**

***Index Terms*—Network Music Performance, Telepresence, Augmented Reality, 5G-NSA, 5G-SA.**

## I. Introduction

Remote human to human interaction is entering the new era of *Augmented Reality* (AR) and *Virtual Reality* (VR), which can provide immersive telepresence based on technologies such as holographic communication. 5G networks with their low latency, high bandwidth and processing at the edge are ideal for such applications. *Network Music Performance* (NMP), where musicians perform together over the Internet, is an ideal use case for 5G. First, NMP can only be effective if the audio latency does not exceed 30–40 ms [1], which is hard to achieve in 4G networks. Second, when more than two participants are involved in NMP, a *Selective Forwarding Unit* (SFU) is needed to mediate between them; the SFU should ideally reside in *Mobile Edge Computing* (MEC) servers close to the participants. Third, while video can greatly enhance NMP, it is immersive telepresence that can completely transform the user experience of NMP participants; however, in addition to ample bandwidth, holographic communication also requires edge processing.

The *Telepresence-Enhanced Network Music Performance* (TENeMP) project[1] is investigating and developing solutions for an immersive NMP experience over 5G. TENeMP is a partner of the SPIRIT project[2], which explores the next generation of telepresence applications in multiple ways, including low-latency networking, scalability to large numbers of users and different forms of telepresence. Crucially, SPIRIT offers *5G Standalone* (5G-SA) testbeds, where the ultra low latency and high bandwidth offered by 5G radios are coupled with a low latency core network and the MEC servers needed to host SFUs and holographic video processing. In contrast, many 5G deployments are actually *5G Non-Standalone* (5G-NSA), combining a 5G radio network with a 4G core network, thus not offering the full set of 5G capabilities.

The TENeMP project is combining audio with video and holographic communication and is investigating the feasibility of different NMP scenarios and topologies over 5G-SA networks. To assess the benefits of 5G-SA, we first need to set a performance baseline, by looking at the behavior of NMP over existing 4G and 5G-NSA networks. To this end, in this paper we provide an overview of our work in TENeMP on the integration of low latency audio, video and holographic communication, as well as our experimental and measurement methodologies. We use these in a video and audio latency measurement campaign over 4G and 5G-NSA networks, to motivate the need to migrate to 5G-SA.

The remainder of this paper is structured as follows. In Section II we review related work. In Section III we describe our experimental testbeds, while in Section IV we present the tools developed by our project. In Section V we show video latency results, from real networks in Athens, Greece, while in Section VI we do the same for audio latency. Section VII concludes the paper and discusses planned future work.

## II. Background and Related Work

Several papers have considered the feasibility of NMP over 5G. Baratè et al. [2] discuss the performance advantages offered by 5G networks to NMP. Centenaro et al. [3] outline a communication architecture for NMP in 5G networks, while Vignati et al. [4] compare NMP performance in 4G and 5G networks through simulations, suggesting that the latter can offer notable gains. There is also some work experimentally evaluating NMP over 5G networks. Dürre et al. [5] conduct an in-depth performance analysis on NMP over a public 5G network in Finland, finding that 5G can support NMP, although performance variability raises concerns. Turchet and Casari [6] explore NMP performance in a private 5G network with four musicians located in the same radio cell, reporting that a continuous stream of reliable, low-latency communication is challenging, even in a private 5G network, and highlighting the need for edge computing. These studies focus solely on audio and do not consider video at all.

[1] https://mmlab-aueb.github.io/tenemp-site/
[2] https://www.spirit-project.eu/

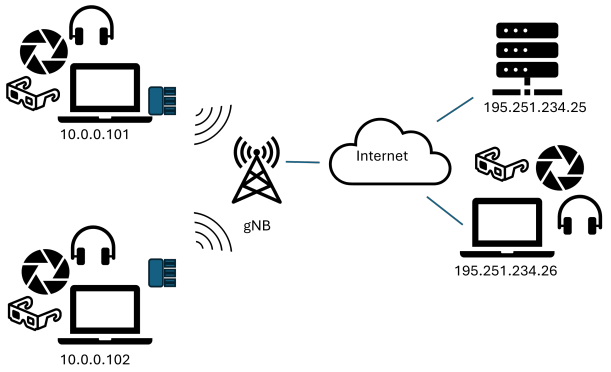


Fig. 1. TENeMP MMlab testbed.

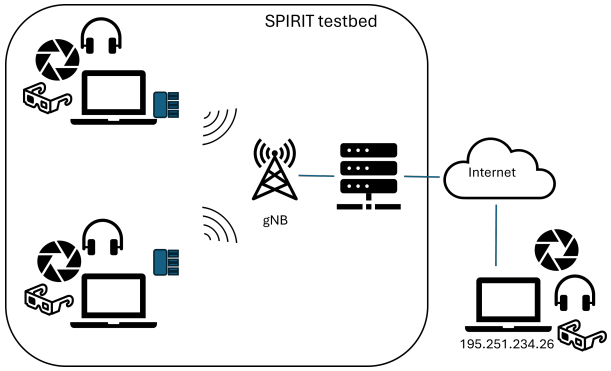


Fig. 2. TENeMP SPIRIT testbed.

Telepresence technologies, such as holograms and avatars, were investigated in the past mostly from a cultural or artistic perspective. K-Live Holographic Music Concerts [7] are an example of using hologram-based arts in traditional spaces for live art production. These events do not currently offer NMP, using instead prerecorded or synthetic performances. The *Avatar Orchestra Metaverse* (AOM) [8] is an example of the popularity of NMP services. AOM is completely virtual, taking place in the Second Life virtual world, with virtual users and instruments. Although there is no media streaming involved, it showcases the cultural impact that NMP can have. Perhaps the only VR study that considers network conditions is by Tamplin et al. [9]; the study explored music performance using VR as a form of therapy for people with spinal cord injury. The setup used a 4G network, while avatar-based telepresence, VR headsets, and different applications were used for audio and VR streams, with roughly 1 s more latency for the VR stream compared to the audio stream.

In contrast to previous work, TENeMP aims to assess the feasibility of immersive NMP over 5G, encompassing not only audio, but also video and holographic telepresence. To achieve this goal, the project will develop integrated NMP tools and assess their performance first over existing 4G and 5G-NSA networks, and then over the SPIRIT 5G-SA testbed.

## III. Local and SPIRIT testbeds

Even though the SPIRIT project offers 5G-SA testbeds, their use for realistic NMP scenarios requires traveling to the 5G sites, which is impractical considering the time needed to test and validate our experimental setup. For this reason, we are running experiments in two different testbeds: first in a local testbed for preliminary experimentation and test validation, and then in a real SPIRIT 5G-SA testbed.

To develop and test the required tools (endpoints and SFU), we have set up a local telepresence testbed at the MMlab in Athens, Greece. The hardware and network setup replicates as much as possible the SPIRIT 5G testbeds, using similar or identical depth cameras (Intel RealSense D435), AR glasses (XREAL Air 2 Pro) and 5G end devices (Samsung S24 5G smartphones and Teltonica RUTX50 5G routers), coupled with low latency audio interfaces (Focusrite Scarlett Gen4) and semi-pro audio sources and sinks. As shown in Figure 1, the testbed consists of two 5G endpoints, as well as a MEC server in the MMlab, with an optional third endpoint reachable over the wired Internet. Each endpoint supports bidirectional ultra low delay audio streaming, as well as volumetric video captured by the depth cameras and replayed by the AR glasses; the local endpoints are either 5G smartphones or laptops connected via Ethernet to a 5G router. Even though our equipment is capable of 5G-SA connectivity, currently only 5G-NSA is available in our area.

Figure 2 shows the corresponding test setup over the SPIRIT infrastructure. The basic component in the testbed is the 5G-SA network, which allows connecting the exact same end devices used at our local testbed with the MEC servers co-located with the 5G network. In addition to complete control over the 5G cell and the provision of 5G-SA services, the actual co-location of the MEC servers with the cell (as opposed to locating them in our lab) allows repeating our experiments in a 5G scenario more amenable to NMP.

## IV. The TENeMP Tools

In the TENeMP project we are conducting experiments to measure end-to-end delay for audio, video and holographic communication in NMP scenarios, under multiple topologies. The topologies relevant for NMP are P2P and Client/Server or, more accurately, *Client/SFU* (CS). While a P2P topology presents the lowest possible latency, when multiple users are involved, the need to send and receive media streams to and from all other participants presents scalability issues. In CS mode, an SFU receives media streams from all participants and replicates and relays them to each participant requesting them, without any processing, to reduce delays. As a result, each participant only needs to send each media stream once.

The hardest part of achieving connectivity between endpoints (and SFUs) is crossing firewalls and NATs. When latency is not critical, the endpoints often resort to indirect communication via a public TCP-based server, as most firewalls allow outgoing TCP connections. For NMP though, this is impractical: connectivity needs to be direct and, preferably, via UDP. One solution to this problem is the WebRTC architecture, which allows endpoints located in different private networks to meet each other and exchange data directly.

WebRTC relies on a set of protocols and services that must be initiated before data exchange can take place. This set

includes the *Session Description Protocol* (SDP), which allows applications to exchange information about the media formats to use, and the *Interactive Connectivity Establishment* (ICE) protocol, which is responsible for exchanging information about the public IPs of the clients and their available ports. The *Session Traversal Utilities for NAT* (STUN) are used to establish a direct UDP connection between two clients; if this fails, *Traversal Using Relay around NAT* (TURN) is used to establish a UDP connection between two clients, which is relayed through a TURN server to bypass firewall rules.


Fig. 3. Bidirectional P2P communication in a LAN.

In a LAN, bidirectional P2P communication, as shown in Figure 3, can be implemented in multiple ways. The Gstreamer framework[3] offers modules for capturing, processing, mixing and creating RTP video and audio streams, which can be connected together to construct pipelines. Another option is the Web APIs[4] which allow JavaScript code to run inside a web browser and access the audio and video devices of the client, as well as various communication protocols.

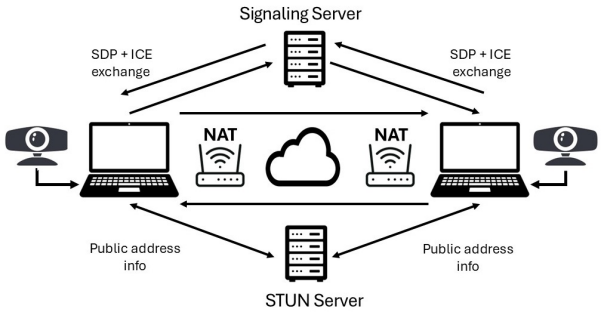


Fig. 4. Peer-to-peer bidirectional video communication behind NATs.

In a WAN, where the endpoints are located in different networks, behind NATs and firewalls, as shown in Figure 4, SDP and ICE information exchange must first take place with a signaling server. The endpoints must find their public IPs and then exchange information about it and their ICE candidates to start the connection. A STUN server is used for the peers to learn their public IP and a signaling server for relaying SDP and ICE messages between them.

While the Web APIs directly support connectivity in such scenarios via WebRTC, they provide limited access to the audio and video parameters which are critical for latency, including the formats used and the buffer sizes. Gstreamer also provides a WebRTC plugin that manages pipelines for video and audio, and additionally emits the SDP and ICE messages required to interface with the signaling server.

For TENeMP we also need to test multiparty scenarios, where an SFU mediates between the endpoints, as shown in Figure 5; the figure omits the NATs and firewalls that the endpoints need to cross to reach the SFU. To develop an SFU, we have considered two options. Jitsi[5] is an open source video-conference project, based on the Web APIs and the WebRTC architecture, providing an SFU, as well as STUN and TURN servers. Although Jitsi is a full solution, it is not really programmable; it is designed to communicate with JavaScript endpoints using the Web APIs. The Janus[6] video relay server is an alternative solution, offering a variety of plugins for WebRTC communication. These can interoperate with a browser-based JavaScript application using the Web APIs, but also with Gstreamer-based standalone clients.

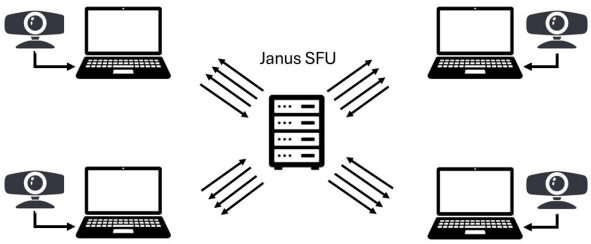


Fig. 5. SFU placement between the NMP endpoints.

## V. Video Latency

*Glass to glass* (G2G) video latency is the time between a single frame being captured by a camera at one endpoint and the corresponding frame being presented at a screen at the other endpoint. Measuring G2G video latency is complicated by the fact that the camera only samples the image at a few tens of Hz and the screen only updates itself at the same rate, but using an independent clock. In addition, video requires compression at the sender and decompression at the receiver, which impose additional delays. As a result, video latency measurements typically exhibit significant variance [10].

In our previous work on NMP, we used a *reflected timer* method to measure latency, with two smartphones located next to each other establishing a video connection [11]. To measure delay, the camera of the first smartphone focuses on a millisecond timer running on a computer monitor. The view of the timer is streamed to the second smartphone and displayed on its screen. By taking snapshots of the monitor and the second smartphone side by side, it is feasible to (manually) calculate the time difference between the original timer and the delayed one on the second smartphone.

Fig. 6. Topology for local LED-based video latency measurements.

[3]https://gstreamer.freedesktop.org/
[4]https://developer.mozilla.org/en-US/docs/Web/API
[5]https://jitsi.org/
[6]https://janus.conf.meetecho.com/index.html

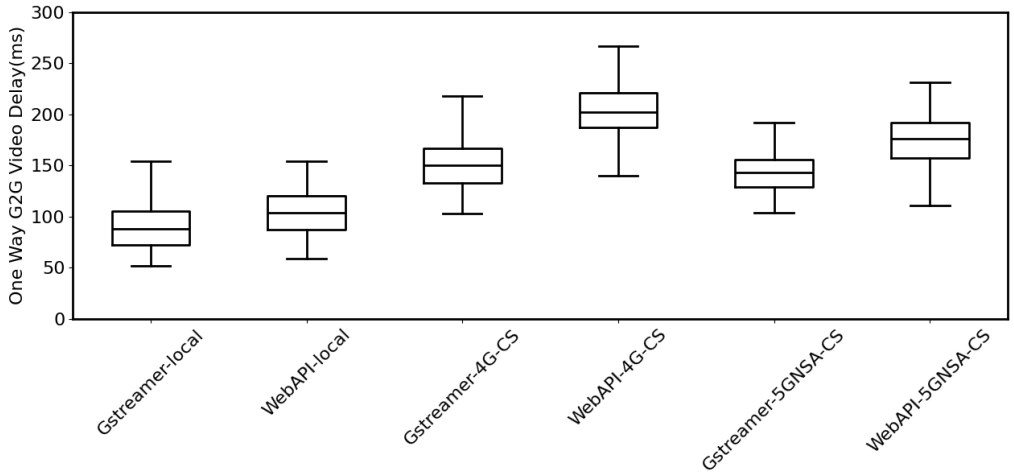


Fig. 7. Comparison of G2G video delay between Gstreamer and WebAPI.

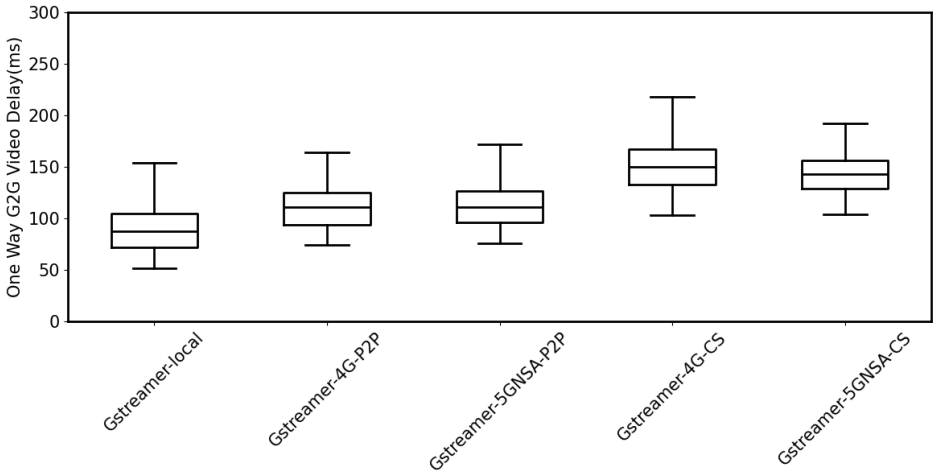


Fig. 8. Comparison of G2G video delay between different topologies, when using Gstreamer.

As this method required manual analysis of the videos to measure latency, in TENeMP we adopted an automated method, based on [10]. We connected a LED to the output of a micro-controller and programmed the LED to turn on for 100 ms and off for 2000 ms. We set a camera connected to a laptop to capture the LED and display the captured image on its screen. We then placed a light sensor facing the laptop's screen and connected it to the micro-controller. We finally programmed the microcontroller to count the interval between the pulse that turns on the LED and the time that the light sensor detected light on the screen, as shown in Figure 6. To measure the additional delay induced by the communication between two endpoints, we sent the captured video to a second laptop located next to the first one, and pointed the light sensor to the second laptop's screen. This allows measuring the G2G video latency with very high accuracy.

To establish a set of G2G latency baselines when using either our Gstreamer or our WebAPI tools we used this method with different topologies, in one-minute trials. We first measured the local G2G latency between the camera and the screen of an Ubuntu laptop; the latencies are shown as the first two boxplots in the Figure 7. We then added a second laptop to receive the video, and connected each endpoint to a separate RUTX50 router (via Ethernet) in 4G only mode. The two laptops communicated via an SFU (Janus) located in the MMlab LAN; the latencies are the next two boxplots. Note that in this setup, the signals had to travel to and from the MMlab LAN. Finally, we enabled 5G mode in the RUTX50 routers, which in our location used 5G-NSA links, and repeated the tests; the results are the final two boxplots.

The boxplots show the median latency (the line inside the box), while the box edges show the 1st and 3rd quartile of the latencies; whiskers show the extent of the latencies that are no more than 1.5 times more than the IQR (the distance between the 1st and 3rd quartile) from the edges of the box. From the boxplots we can see that Gstreamer has a consistently lower G2G latency than WebAPI; this is more evident in the SFU tests. We can also see that the baseline G2G latency is quite high even for a local setup, indicating the need to optimize the capture/compression/playback pipeline. Finally, we can see

that replacing 4G with 5G-NSA does lead to a reduction in G2G latency, although the results are not spectacular.

As Gstreamer was shown to induce lower delay than the WebAPIs, we conducted a second set of trials to compare the G2G latency between the P2P and SFU topologies; the results are shown as boxplots in Figure 8, where we also show the local G2G latency (camera to screen) for reference. In P2P mode we tested both laptops on 4G links and then on 5G-NSA links, and then used the same topologies but in SFU mode. It is interesting to note that in the P2P topologies the latencies are quite close with all link types, but when we add an SFU (recall that the SFU is located in the MMlab), the delays grow a lot, indicating the potential benefits of having the SFU at a MEC server in the cell.

## VI. Audio latency

*Mouth to Ear* (M2E) audio latency is the time between a user producing a sound and the sound reaching the ears of another user. We have previously conducted M2E measurements with various audio tools using a reflected pulse method [12]. In this approach, we produce audio pulses, which are sent to a receiver, played back there, captured again and returned to the sender; we record both the original and the returned signal as the left and right channels of a stereo signal, and then visually inspect the recorded waveforms to calculate the round trip delay of the signal. Assuming a symmetric connection, half of that delay is the M2E audio latency.

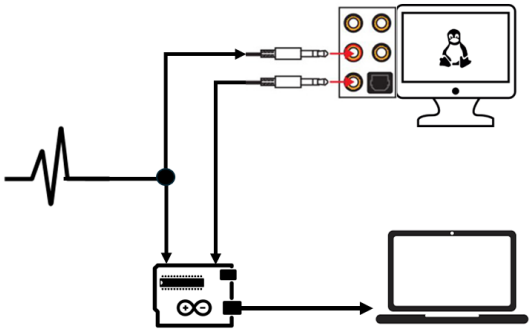

Fig. 9. Topology for local pulse-based audio latency measurements.

To automate the measurements and directly capture M2E, rather than round trip, latency, for TENeMP we used a slightly modified method. We first use Audacity[7] to periodically generate short audio pulses every 1.8 sec. We then split the audio signal, sending one copy to the a computer's line input, and the other to a microcontroller's analog input. The computer captures the input audio and plays it back, and we connect its line output to a second analog input of the microcontroller. The microcontroller then calculates the delay between the two pulses, as shown in Figure 9. This allows us to capture the M2E delay due to local capture and playback operations. To measure the additional delay induced by the communication between two endpoints, we send the captured audio to a second laptop, located next to the first one, and compare the line output from the second laptop with the line input sent to the first laptop, using the exact same method.

Unlike video, where no NMP-optimized tools exist, for audio two well-known NMP-optimized tools are Jacktrip[8] and Sonobus[9]. Jacktrip is an ultra-low audio delay tool optimized for very fast audio streaming in P2P mode, but it has issues with NAT traversal. For this reason, we combined it with Sonobus which supports NAT traversal. Specifically, audio was captured by Jacktrip, passed to Sonobus for transmission, Sonobus received it a the other end and passed it to Jacktrip for playback. We also tested our Gstreamer and WebAPI-based tools; we omit resuls from the WebAPI tool due to is high latency, which was as evident as in the video tests.

We executed a number of experiments using the above method, lasting for 3 minutes each (corresponding to around 100 pulses). Audio was sent uncompressed (PCM), sampled at 44.1 KHz with 16 bits per sample. We first measured the local M2E latency between the microphone and the speaker of an Ubuntu laptop; the latencies for the Jacktrip/Sonobus and Gstreamer-based tools are shown as the first two boxplots in Figure 10. We then added a second laptop connected via a LAN as an audio receiver, measuring the P2P M2E audio latency in a LAN setting; these latencies are the next couple of boxplots. Finally, we connected the two laptops to the RUTX50 routers (via Ethernet) in 5G-NSA mode, with the results shown as the last two boxplots. While there is a small latency increase in the LAN seting, the M2E delay is less than 20 ms, which is quite acceptable for NMP. However, with the 5G-NSA links, even without an SFU between the endpoints, the delays grow considerable, to a median of 70 ms for Jacktrip/Sonobus but only slightly over 40 ms for Gstreamer, which is borderline acceptable for NMP.

## VII. Conclusions and Future Work

5G networks promise ultra low delay, high bandwidth and processing close to the mobile edge. These aspects could not only make NMP, with its high sensitivity to latency, a reality, they could also transform NMP into an immersive experience, by coupling audio communication with advanced telepresence technologies, such as holographic communication. The TENeMP project is developing NMP tools that will be tested over the SPIRIT 5G-SA testbeds, specifically targeting those aspects of 5G-SA which can enable immersive telepresence.

In this paper we described our local 4G and 5G-NSA testbed, as well as the tools that we have developed to provide audio and video connectivity to NMP participants. We then presented some initial latency results for video and audio transmission over 4G and 5G-NSA connections, using different tools. The goals of these tests were to select the most appropriate tools, gather baseline measurements from 4G and 5G-NSA environments to assess the possible benefits of 5G-SA and, finally, test the measurement methodologies that will be used for the experiments in the SPIRIT 5G testbed.

We expect the initial results shown in this paper to be improved when testing over 5G-SA links, due their lower latency

[7]https://www.audacityteam.org/

[8]https://ccrma.stanford.edu/software/jacktrip/

[9]https://sonobus.net/

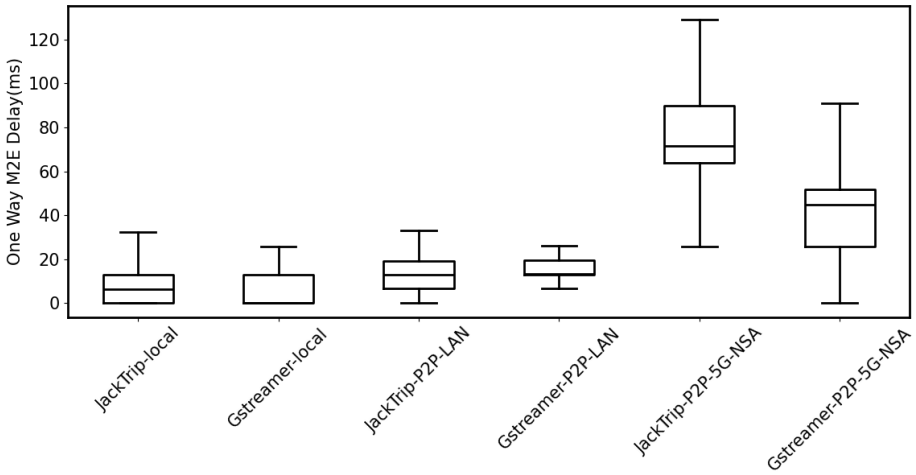


Fig. 10. Comparison of M2E delays using Jacktrip-Sonobus and Gstreamer.

compared to 5G-NSA; in the SPIRIT 5G testbed we also expect latency improvements in the SFU configurations, since the SFU will be placed in MEC servers close to the endpoints. Furthermore, we have not yet optimized the Gstreamer video pipeline, which currently consists of grabbing raw frames and compressing them with vp8 in low latency mode at the sender; we are evaluating alternative pipelines to minimize the (de)compression latency of video, including lower latency compression schemes (like MJPEG) and compression in the camera. In addition, we are researching low-latency methods of getting video and volumetric information from the depth cameras, which offer an extensive programming API. Finally, we have only experimented with video and audio so far; we are currently working on adapting our G2G latency measurement setup for holographic communication, by projecting the received volumetric video to a (2D) screen and reusing our LED-based automated latency measurement scheme.

## ACKNOWLEDGMENT

The work reported in this paper has been partly funded by the EU through the subgrant Telepresence-Enhanced Network Music Performance (TENeMP, SPIRIT-OC1) of project SPIRIT (grant agreement No. 101070672). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the EU. Neither the EU nor the granting authority can be held responsible for them. The SPIRIT project has received funding from the Swiss State Secretariat for Education, Research and Innovation (SERI).

## REFERENCES

[1] K. Tsioutas, G. Xylomenos, and I. Doumanis, "An empirical evaluation of QoME for NMP," in *IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, 2021, pp. 1–5.

[2] A. Baratè, G. Haus, and L. Ludovico, "Advanced experience of music through 5G technologies," *IOP Conference Series: Materials Science and Engineering*, vol. 364, p. 012021, 06 2018.

[3] M. Centenaro, P. Casari, and L. Turchet, "Towards a 5G communication architecture for the Internet of musical things," in *27th Conference of Open Innovations Association (FRUCT)*, 2020, pp. 38–45.

[4] L. Vignati, G. Nardini, M. Centenaro, P. Casari, S. Lagén, B. Bojovic, S. Zambon, and L. Turchet, "Is music in the air? evaluating 4G and 5G support for the Internet of musical things," *IEEE Access*, 2024.

[5] J. Dürre, N. Werner, S. Hämäläinen, O. Lindfors, J. Koistinen, M. Saarenmaa, and R. Hupke, "In-depth latency and reliability analysis of a networked music performance over public 5G infrastructure," in *Audio Engineering Society Convention 153*, 2022.

[6] L. Turchet and P. Casari, "Latency and reliability analysis of a 5G-enabled Internet of musical things system," *IEEE Internet of Things Journal*, vol. 11, no. 1, pp. 1228–1240, 2023.

[7] W. J. Chang and H.-D. Shin, "Virtual experience in the performing arts: K-live hologram music concerts," *Popular Entertainment Studies*, vol. 10, no. 1-2, pp. 34–50, 2019.

[8] G. Martín, "Social and psychological impact of musical collective creative processes in virtual environments; the avatar orchestra metaverse in second life," *Music Technology*, vol. 75, pp. 75–87, 2018.

[9] J. Tamplin, B. Loveridge, K. Clarke, Y. Li, and D. J Berlowitz, "Development and feasibility testing of an online virtual reality platform for delivering therapeutic group singing interventions for people living with spinal cord injury," *Journal of telemedicine and telecare*, vol. 26, no. 6, pp. 365–375, 2020.

[10] C. Bachhuber and E. Steinbach, "A system for high precision glass-to-glass delay measurements in video communication," in *IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 2132–2136.

[11] K. Tsioutas and G. Xylomenos, "Assessing the effects of delay to NMP via audio analysis," *SN Computer Science*, vol. 4, 2022.

[12] ——, "Audio delay in web conference tools," in *Workshop on Web Engineering and Collaborative Music Learning (WECML)*, 2022, pp. 1–6.