benchmarking networked music performance tools with the nmp-bench model and architecture

Benchmarking Networked Music Performance Tools with the NMP-Bench Model and Architecture

Matteo Gori
University of Florence
Florence, Italy
mat.gori15@gmail.com

Andrea Ceccarelli
University of Florence
Florence, Italy
andrea.ceccarelli@unifi.it

***Abstract*—Networked Music Performance (NMP) aims at establishing a live interaction between musicians remotely connected, which perform as if they were in the same room. While several NMP tools have been proposed through the last 20 years, a benchmark that aims at measuring and comparing their quality is currently not present. In this paper we propose the NMP-Bench (Networked Music Performance Benchmark) benchmark, the first approach to systematically analyze and compare the performance of NMP tools. Focused on server-based NMP and its auditory component, NMP-Bench provides a comprehensive approach to measure and quantitatively compare NMP tools, encompassing network, music, and effectiveness metrics, with the goal of understanding the technical gaps that may reduce the experience of the performers. The paper presents the NMP-Bench model and architecture, which is then applied to benchmark two NMP tools (in simulated settings with no actual musicians) over three music pieces of different music styles. Results show differences between the two tools; this supports our statement that NMP-Bench can be used to select the most suitable tool and to highlight strengths and weaknesses of NMP tools.**

Keywords— Networked Music Performance, Quality of Service, music, benchmarking, IoT applications.

I. INTRODUCTION

Nowadays, Networked Music Performance (NMP, [8]) represents the most promising expression of musical interaction through the network, such that it has been recently identified as one of the fundamentals of the Internet of the Musical Things [36]. NMPs may involve conventional musical instruments or, recently, the family of smart musical instruments [2], which are characterized by wireless connectivity to local networks and to the Internet.

More practically, NMP makes it possible to establish a live interaction on a network between musicians located in different places, aiming them to perform as if they were in the same room [1-2]. Several tools for Networked Musing Performance have been proposed in the literature over the last 20 years, just to name a few Soundjack [17], Jamulus [16], NINJAM [18], MusiNet [19], Jamberry [20]. All of them have to devise technical solutions to overcome the several limiting factors to a successful NMP experience. Such limiting factors originate from the nature itself of asynchronous distributed systems [34], and ultimately manifest in an unpleasant experience for both the performers and the listeners. Examples of such factors are unreliable network communications, latency, jitter, clocks offset and drift, nodes with low processing capacity.

Analysing the state of the art, we observe that NMP tools are evaluated by their authors, often with extensive experimental campaigns. However, typically each paper follows a different evaluation strategy; an univocal approach to the evaluation of

NMP tools is currently missing, that would provide an objective understanding of their quality and would facilitate comparison of repeatable evaluation results. In other words, a benchmark for NMP tools is currently not available. Benchmarks are standard procedures that allow evaluating and comparing different systems, components and tools according to specific characteristics [15], and that have been largely used to compare the performance of systems as for example web services, networks, or transactional systems.

This paper presents NMP-Bench (*Network Music Performance Benchmark*), the first proposal of a benchmark for server-based NMP. NMP-Bench is organized in two parts: the NMP-Bench model and the NMP-Bench architecture. The first part is the definition of a general model where relevant quantities for NMP benchmarking are collected, through literature review of relevant existing works in the domain. Attributes and metrics have been selected including network, music and effectiveness attributes, which put the focus on the auditory component of the performance. This is the fundamental part of NMP-Bench, as it is well-known that benchmarking can facilitate the comparison of tools, but its effectiveness is strictly dependent on the adequacy of the attributes to evaluate [15]. As it will be evident during the paper, we devoted particular attention to describe the experience of the performer rather than the listener, because nowadays NMP tools are mostly intended for the temporary and sporadic aggregation of music band for recreative purposes, rather than for high quality music listening. The second part is the definition of the NMP-Bench architecture, in which we propose a possible measuring instrument and measurement process for NMP benchmarking.

The rest of the paper is organized as follows. Section II provides a general overview of NMP, with basics and motivations of our work. The definition of the NMP-Bench model is in Section III: it shows the various metrics that constitute the evaluation scheme of our benchmark. The NMP-Bench architecture is in Section IV, with instructions on the process to exercise NMP-Bench on NMP tools. Section V presents the instantiation, execution and results of NMP-Bench on two NMP tools. Finally, we elaborate on limitations of our study in Section VI, while concluding remarks and future works are in Section VII.

II. Background and State of the Art

A. Basics on Networked Music Performance

The term Networked Music Performance (NMP) is known since many years, as it is shown in [1] where it is dated back to at least 1978. Despite countless steps forward in the technological domain, still nowadays NMP is facing structural and development problems, that impacts the quality of the solutions available. NMP solutions have to face, amongst other:

i) *clock (un)synchronization*. It is evident that tight clock synchronization of NMP clients is mandatory. Solutions for clock synchronization of distributed nodes are well-known since long [3], although a certain synchronization error is unavoidable.

ii) *packet loss*. It is one of the most delicate aspects when managing a network. Packet loss is problematic during an NMP session, since a normal sound stream is a mandatory requirement for an audio transmission. Multiple cases can lead to packet losses. Among them, we name transmission errors, network congestions, node or connection losses. The re-transmission of lost packets is not well-suited for real-time interactive applications as NMP, consequently loss-recovery schemas for audio streams and error-concealment methods have been studied through the years [4-6] and are still researched nowadays [38-39].

iii) *network latency*. It is probably the major problem for NMP. In this paper we refer to two different latencies: the OOSE (Over-all One-way Source-to-Ear) and the RTT (Round-Trip Time). The OOSE is a one-way delay that includes all the cumulative delays between the acquisition of the incoming audio signal and its output playback [8]. Instead, RTT is simply the time required by a packet to travel from a given source to a remote destination and go back.

iv) *jitter*. It is the variation of one or more characteristics of a signal from a point of origin to its destination. For audio streams, jitter is often mitigated by applying playback delays (or playout delays) to the various packets when they are received [7].

B. State of the Art on Benchmarking NMP Tools

In the recent scientific literature, we can find numerous examples of excellent experimental studies on NMP tools. Authors of these works usually present an exhaustive analysis of an individual NMP tool or on tool internals.

In this paper, we observe that such evaluations generally exploit methodologies that are devised by the authors themselves, and consequently inconsistencies between different works (especially when from different authors) are unavoidable. Further, it is possible that some evaluations lack completeness e.g., forgetting to measure relevant metrics for NMP. In other words, if we consider the task of benchmarking NMP tools to produce comparable results that are collected through a reproducible evaluation procedure, the situation is significantly different and, to the best of our knowledge, without any relevant example.

Further, it should be observed that surveys of NMP technologies and tools exist, for example [8], [9], [10]. Especially, [8] is a main reference for our work as it discusses relevant metrics to assess NMP tools. However, all these works do not identify a framework for the evaluation of NMP tools, but rather such tools are compared on the basis of information collected from the works of the various authors.

This work was conceived, studied and developed to fill such gap in NMP tools assessment. The NMP-Bench (Network Music Performance Benchmark) model and architecture for benchmarking NMP tools is comprehensive of all the main technical disciplines involved in evaluating NMP tools, including network, music, and usability. We believe NMP-Bench is the first step to fill the assessment gap that is still open in this discipline.

III. Building the NMP-Bench Model

We define the NMP-Bench model that contains all the attributes and metrics required to apply the NMP-Bench benchmark. Using a bottom-up approach, we first introduce the attributes and metrics that compose the model. These are grouped in i) network performance in Section III.A, ii) musical scope in Section III.B, iii) effectiveness attributes in Section III.C, and iv) user satisfaction in Section III.D. Most of the attributes and metrics are collected through literature review, and mainly from [8], [26], [28]. Finally, in Section III.E we organize the various metrics to present the entire NMP-Bench model.

A. Networks Performance Attributes and Metrics

When referring to the performance of a network, it is necessary to consider Quality of Service (QoS). The attributes that we identify to describe network QoS for NMP tools are responsiveness, efficiency, packet loss, productivity; these are studied and classified in the following. For each attribute, one or more metrics are proposed to measure it. When relevant, a range of plausible values is matched to a metric; the scope of these values is to provide plausibility intervals among which measured values are expected to fall. Values outside the plausibility interval may suggest the NMP-Bench user to verify data and the measuring instrument. Metrics and their plausible values are collected from literature, mainly from [8].

Responsiveness. It refers to the specific capacity of the whole system or of a single functional unit to complete the assigned tasks within a given time [14]. Responsiveness is a fundamental attributes for any NMP solution. We measure this attribute with two metrics, namely the delay OOSE d_{OOSE} and the round-trip time *RTT*. Concerning the RTT, it is a well-known metric typically computed by network sniffers; how to compute RTT is not discussed here, and interested readers may refer to [7]. Instead, the computation of the d_{OOSE} of an NMP system is from [8] and it is described with the help of Table I (see rows corresponding to metric d_{OOSE}). It is:

$$d_{\mathrm{OOSE}} = (N \cdot d_{\mathrm{soundcard}}) + d_{\mathrm{buffer}} + d_{\mathrm{network}} \qquad (1)$$

where:

- N represents the number of connections,

TABLE I. Metrics and plausible values that support the computation of d_{OOSE}, $Ch_{\mathrm{utilization}}$, THR.

Metrics		Description	Plausible Values
d_{OOSE}	N	Number of connections	2-16 (from our experience)
	P	Sound card block size	64-512 samples [8]
	R	Sampling rate	8-96 kHz [8]
	B	Application buffer size	2-16 blocks [8]
	C	NIC bandwidth	0.054-20 Mbps [8]
	Ch	Number of application audio channels	1-16 [8]
	O	Total overhead of a data packet	< 1000 bits [8]
	δ	Sampling depth of the sound card	16-24 bits/sample [12]
	σ	Codec compression ratio	0.25-1 [8]
$Ch_{\mathrm{utilization}}$	br	Link transmission speed	v/s
	L	Packet size	byte
THR	$RWIN$	Size of the receiving window	≤ 65 kbit (TCP protocol [21])

- $d_{soundcard}$ is the sound card delay. It is computed as $d_{soundcard} = d_{sampling} + d_{encoding/\ decoding} + d_{blocking}$. We have that $d_{sampling} = 1/R$, with R the sampling rate of the audio signal in hertz; $d_{encoding/decoding}$ is considered only if the software examined uses an audio codec for the compression of the audio signal; finally, $d_{blocking} = P/R$, where P are the samples to be generated before the processors start to recover the audio signal.
- d_{buffer} is the application buffer delay, and it is:

$$d_{buffer} = (B\ P) / (N\ R)$$

where B is the block size of the application buffer.

- $d_{network}$ is the network delay, which is $d_{network} = d_{processing} + d_{propagation} + d_{transmission}$. We have that $d_{processing}$ is determined by multiple factors including the network topology and routing algorithms, and $d_{propagation}$is a propagation delay that depends on the transmission medium. Instead $d_{transmission} = R\ \sigma\ Ch\ (\delta + O\ \delta) / P\ C$, where σ quantifies the codec compression ratio, Ch indicates the number of audio channels of the application, δ denotes the sampling depth of the sound card resolution in bits/sample, O shows the total overhead of a data packet in bits and finally C designates the bandwidth of the NIC (Network Interface Card) in Mbps.

Efficiency. The key metric for the attribute efficiency is channel utilization, that measures the time to transmit the data versus the total time to transmission frames including any overhead to make use of the channel. In its general formulation, channel utilization with $Ch_{utilization}$ is a percentage, and it can be computed as [7, 13]:

$$Ch_{utilization} = t_{transmit} / (t_{transmit} + t_{ACK} + (2 \cdot d_{propagation})) \qquad (2)$$

where (see also Table I):

- $d_{propagation} = D/V$, in which D denotes the length of the channel in kilometers (km) and V identifies the propagation speed in m/s.
- $t_{transmit} = L/br$, where L indicates the length of the frame in bytes and br is the bit-rate in *kbps* (kilo-bits per second).

Two observations are necessary on equation 2. The first one is that in the Internet Protocol Stack, the ACK is needed only for the TPC protocol (which is connection-oriented) and consequently for applications implemented on top of TCP, The contribution of the ACK shouldn't be considered for the connectionless alternative UDP and consequently for (streaming) applications that typically exploits UDP.

Second, in many cases the time t_{ACK} to generate the ACK is not considered at all. This is because the ACK message is much shorter than the transmitted message, and its generation time is negligible with respect to $t_{transmit}$ and $d_{propagation}$, so it is not relevant in Equation (2).

Packet loss. A common metric is the Packet Loss Rate *PLR* i.e., the percentage of frames that should have been forwarded through a network but that never reached the desired destination [11].

Productivity. The last attribute that we present is the productivity of a network. The main metrics that allow to measure productivity are the two well-known bandwidth and throughput. Bandwidth *BW* is the maximum number of bits that can flow through a network connection over a period of time; its value is usually computed as $BW = CPS / RCT$, where *CPS* is the cumulative packets size that indicates the sum of the size of the various packets sent, and *RCT* is the relative capture time that denotes the time taken by the system to capture all the packets sent. Throughput *THR* is the quantity of data successfully transferred from one place to another through a certain communication channel in a given period of time. For example, for TCP the throughput can be calculated as $THR \leq RWIN / RTT$, where *RTT* is the Round-Trip Time and *RWIN* identifies the receive window, that is, the amount of unacknowledged data which TCP can send to keep the pipeline full [21] (see also Table I).

B. Musical Scope Attributes and Metrics

Our discussion cannot, for obvious reasons, address all the various elements (melody, notes, sound, etc.) that constitute the musical universe. Instead, we will select only those elements that we consider important for the evaluation of NMP tools from the perspective of the performer, for the reasons already explained in Section I. Consequently, we present here the metronomic attribute.

Metronomic allows specifying the rhythmic scanning of a musical sequence through the use of the metronome. Metronomic can be measured using metrics that are based on the beats per minute (bpm), the measurement unit that allows obtaining the speed of a track. We describe metronomic using the metrics maximum and minimum peak of bpm in a song (maximum bpm and minimum bpm), the average between the various bpm of the music track taken into consideration (average bpm), and finally the trend, defined as the bpm value that appears most frequently in the music track (trend bpm).

The calculation of the four metrics presented can be easily performed through a bpm-counting software, and does not require further explanation. However, it is necessary to define an input parameter that significantly influences the results. This is the size of the time windows *TW* in which the music track is split. In fact, *TW* segments of a few seconds (from our direct experience, less than 10 seconds) make it possible to obtain a high degree of precision on the calculation of the proposed metrics, given the fact that the bpm is measured over several small segments. Noteworthy, large windows *TW* (from our direct experience, above 25 seconds) significantly simplify the computation, since a smaller number of segments is produced, but inevitably introduce greater accuracy errors.

C. Effectiveness Attributes and Metrics

We refer to effectiveness as the accuracy and completeness with which users achieve specified goals [40]. To evaluate effectiveness, we select attributes that can be measured quantitatively to objectively comment on the quality of the performance in terms of performer's experience. Specifically, we introduce the attributes time and accuracy.

Time. Time is the speed or the rhythm of a given track; it is one of the pillars of the musical universe. Having the rhythm disrupted is generally the most severe concern for NMP users (and musicians in general). A study of time is therefore essential when analysing the user satisfaction in using an NMP tool.

Time is organized in pacing and slope. Pacing is the average speed maintained by an interpreter during the reproduction of a track in any form of interaction with other musicians [26]. This result is very important to verify the quality of a musician's performance with respect to the original track. Given *A* one musician, *W* the number of time windows, and BPM_i^A the average *bpm* of the *i*th time window for the musician *A*, it is possible to calculate the pacing Π_A as [8]:

$$\Pi_A = \frac{1}{W}\sum_{i=1}^{W} BPM_i^A$$

Instead, slope describes how the pacing of a musical piece changes over time. In other words, it detects if the musician *A* has accelerated or decelerated its pace during his performance. Given $Average_{BPM}$ the average BPM for a music track, we measure the slope *k* for the musician *A* in the following way: $k_A = Average_{BPM} - \prod_A$.

Given k_{TRACK} the slope from the sheet music i.e., the slope measured when the song is played perfectly, it is easy to observe that:

a) $k_A = k_{TRACK}$ means the musician *A* performed the song in an optimal way;

b) $k_A > k_{TRACK}$ means the musician *A* accelerated compared to the sheet music, and

c) $k_A < k_{TRACK}$ means the musician decelerated.

In the NMP-Bench model (in Table II), pacing and slope are calculated as difference from their values computed using the sheet music (that is, the ideal values when the song is played perfectly) and the pacing and slope of the music played by the musician during an NMP session. In other words, pacing and slope express the distance in bpm from the intended performance and the actual performance.

Obviously, pacing and slope have to be computed for each musician individually, and they represent the individual experience of a musician in terms of capability to adequately keep the rhythm. Although pacing and slope depend also on the quality of the musician and the difficulty of a song, it is reasonable to assume that, as long as musicians are professionals and the music tracks are the same, their measurement using different NMP tools can be performed under reproducible conditions [27], and consequently measurement results are comparable.

Accuracy. Accuracy usually indicates the closeness of agreement between a measured quantity value and the true quantity value of a measurand [27]. Translating this definition to NMP assessment, the values from the sheet music are the true quantities, and the values measured during the performance are the quantity values. We introduce three metrics that allow quantifying accuracy of a musical performance: asymmetry, imprecision and regularity.

In fact, asymmetry allows checking the average execution delay of a musician *B* compared to another musician *A* [35]. Denoting $\mathrm{ASY}_{\mathrm{AB}}$ the asymmetry between the two musicians, it is possible to compute asymmetry through the following formula [8]:

$$ASY_{\mathrm{AB}} = \frac{1}{W}\sum_{i=1}^{W} |BPM_i^A - BPM_i^B|$$

Instead, imprecision analyses the variability of the execution time between the musicians A and B involved in the performance, that is the standard deviation of the differences in the performance times (in bpm) of the two users [28]. This calculation leads to the evaluation of a single imprecision, that is the one obtained between two interpreters. In most performances, however, musicians are not limited to a duo. We define the imprecision ensemble as the average of the individual inaccuracies that have developed between the various performers involved in the interaction [28]. For example, supposing a musical performance between three musicians *A*, *B* and *C*, the calculation of the ensemble imprecision *SD* is the average between the individual inaccuracies detected among the various musicians i.e., $SD = (SD_{\mathrm{AB}}+SD_{\mathrm{AC}}+SD_{\mathrm{BC}}) / 3$.

The third and final metric is regularity. It measures the imprecision of timing within the entire track performed by a musician *A* [26]. Its value is expressed in percentage terms (percentage of uniformity), and it is measured as the coefficient of variation $CV = SD_{\mathrm{A}} / M$, where SD_{A} is the standard deviation calculated with respect to the average bpm of the music track *M* (obviously, $M \neq 0$ always holds) for a single musician *A*. This metric assumes values in the interval [0, 1]; a low value means the musician was able to perform according to the sheet music.

D. User Satisfaction

We complete the discussion by briefly mentioning user satisfaction i.e., the degree to which user needs are satisfied when a product or system is used in a specified context of use [40]. User satisfaction of a musical interaction is a complex psychological aspect that may be evaluated by a subjective approach, to collect and synthesize the individual sensations of the various users involved in the interaction as a result of their experience. User satisfaction depends on a variety of factors that can be measured with multiple tools. Among these, the most used are certainly short questionnaires that are submitted to users immediately after the conclusion of the experience, which in our case is an NMP session. For example, we can identify template questionnaires as After Scenario Questionnaire [23], Single Ease Question [24], NASA Task Load Index (NASA-TLX) [25]. Such surveys are carried out once the NMP session has ended and they allow analyzing if the musicians achieved a satisfactorily music performance. In other words, they allow verifying the ability of the various musicians to keep the fluidity and the contemporaneity of the interaction as much as possible in terms of time.

Adapting these questionnaire to NMP tools and experience is straightforward, and a candidate questionnaire is not reported here for brevity. Although a questionnaire is very important to fully understand the degree of usability of a system, it is deemed secondary within this paper, which focuses on presenting the overall NMP-Bench benchmark.

E. The NMP-Bench Model

The NMP-Bench (Network Music Performance Benchmark) model is reported in Table II. It is organized in three different areas, one for each field of study: network performance, musical scope, user satisfaction. To the best of our knowledge, this model is the first one to cover these three fields. The various attributes for each field, as presented during this Section, are reported in Table II together with the metrics that allows quantifying them. When applicable, plausible values are also reported. Finally, measurement units for the selected metrics are included. The simplicity, the applicability and the ease of reading and computation are strengths in favor of this scheme.

As mentioned in Section III.D, questionnaire are not reported but they complement this scheme for what concerns user satisfaction.

TABLE II. STRUCTURE OF THE NMP-BENCH MODEL.

Attribute	**Metrics**	**Plausible values**	**Measurement unit**
	Network performance quantities and metrics		
Responsiveness	$delay_{OOSE}$ RTT		ms ms
Efficiency	Channel utilization	≤ 88 *(construed from Table I)*	% of busy time
Packet loss	Packet loss rate	≤ 1	% of packets loss
Productivity	Bandwidth Throughput		Mbps
	Musical scope quantities and metrics		
Metronomic	Maximum BPM Average BPM Minimum BPM BPM trend	*plausible values may come from classifications of tempos and genres*	bpm bpm bpm bpm
	Effectiveness quantities and metrics		
Time	Pacing Slope	± 4 [33] ± 4 [33]	bpm bpm
Accuracy	Asymmetry Single imprec. Ensemble imprec. Regularity	± 4 [33] ± 4 [33] ± 4 [33] ± 4 % [33]	bpm bpm bpm % of uniformity

IV. NMP-BENCH ARCHITECTURE

This Section explains the possible design of a measuring system to collect data and measure the NMP-Bench attributes and metrics, and the steps for a proper measurement process. However, such architectural design should be interpreted with higher flexibility with respect to the NMP-Bench model, because some parts may require different instantiations or settings depending on the characteristics of the NMP tool under study. It should be remarked that the objective is to build a measuring system, and as such, any measuring instrument can be deemed adequate as long as it guarantees low intrusiveness (the measuring instrument itself does not alter results), high measurement repeatability, low measurement uncertainty, and appropriate resolution, amongst main metrological properties [30], [27].

The general architectural schema that we propose is intended for NMP architectures that are server-based i.e., an NMP server is in charge of receiving the different tracks from the musicians and coordinating the whole performance. However, the definition of an analogous NMP-Bench architecture for peer-to-peer NMP tools is straightforward starting from the information reported here. Several NMP tools relies on a server-based approach, for example TransJam [29], Jamulus [16], MusiNet [19], NINJAM [18].

We describe the NMP-Bench architecture with the help of Figure 1, from right to left. In the rightmost part of the figure we find the NMP application server: it is a network server that has an NMP application installed. The NMP application server conveys all the audio data transmitted by the various musicians and redistributes them, guaranteeing a synchronous performance in the best possible way. Audio data are exchanged between the musician PCs and the server for the entire duration of the interaction.

The rest of Figure 1 includes components that runs locally on the musician PC. Each musician interested in collecting the relevant data and perform measurements according to the NMP-Bench model should have a network monitor installed on such PC. In fact, a network monitor is a traditional, easy, and appropriate approach to compute network performance quantities. There are several network sniffing tools available for the main Operating Systems, that can observe and record audio packets in both directions. Noteworthy, intrusiveness of such network sniffer should be estimated, and data managed accordingly. Several

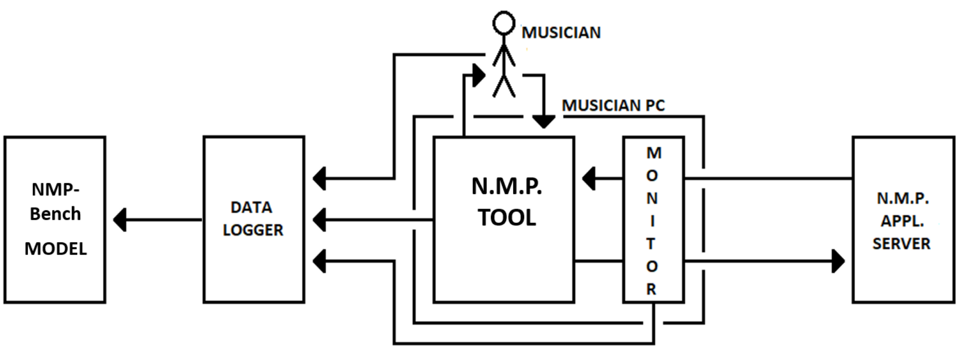

Figure 1. A possible NMP-Bench architecture, to instantiate the measuring system for benchmarking.

approaches to measure, reduce or mitigate intrusiveness of network instruments are available [30], but they are not debated here; it is generally required that the measurement system is devised and instantiated according to principles from measurement theory [27].

The NMP tool in Figure 1 is the target of the benchmarking activity. It is used by the musician. The music tracks that are produced (and that may be transmitted to the server, depending on the kind of NMP tool) are recorded as well, to compute musical scope quantities and user satisfaction quantities. NMP tools may offer functionalities to record their audio outputs; alternatively, outputs of audio streams can be easily recorded, as several tools are available for the main Operating Systems.

Finally, we discuss the boxes data logger and NMP-Bench model in Figure 1. These are intended to store data and perform offline data processing. All the information collected during the music performance, that comes from the network, the musician, and the NMP tool, is logged and then elaborated to compute all the quantities of the NMP-Bench model in Table I. The simplest implementation is to i) locate the logger on the musician PC, and ii) perform data analysis on the collected data, directly working on such PC. Alternatives, not explored here for brevity, require the definition of a remote database that collects data from different musicians, and procedures to automate data processing and analysis; an interested reader may refer to [31] for a reference approach although outside the NMP domain.

To further explain data collection and analysis, Figure 2 is presented. The area in the grey box named Computation Phase describes how data is gathered and elaborated to compute the metrics of the NMP-Bench model. We present it from top to bottom. The reference values e.g., the true values of average bpm, pace, slope, are extracted from the studio version of the music track or from sheets music. Once the NMP music performance ends, the musician has also stored the track recorded during the music performance. More precisely, each musician records the track of his own performance i.e., his own instrument. For simplicity, in this discussion we consider the studio music track and the recorded track of only one generic musician *A* playing any instrument.

These two tracks shall be examined to obtain the values of the metronomic attribute; this can be done with any audio analysis tool. The box Comparison of Results computes the user satisfaction of the NMP-Bench model. Still in Figure 2, the Capture File

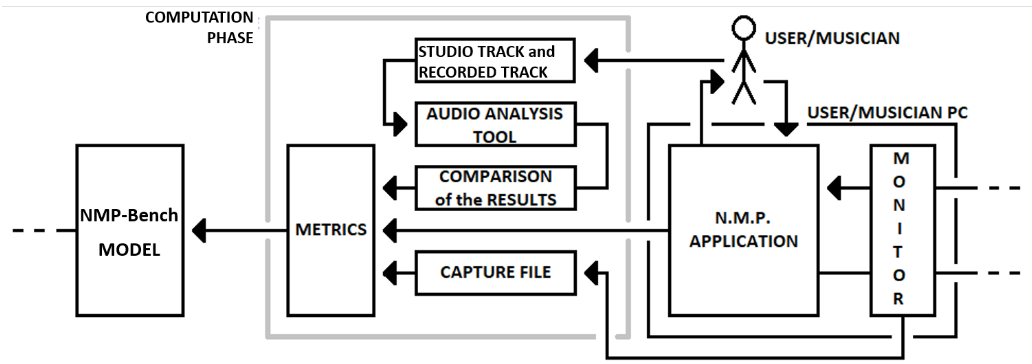

Figure 2. Data collection and processing.

box means that data obtained through the use of the network sniffer is saved, and it allows computing network performance metrics.

To compute metrics related to accuracy, it is necessary that tracks from the different musicians participating to the NMP music performance are collected, and jointly analysed. This is a straightforward adaptation to Figure 2 and not further debated.

The values achieved must be compared with the plausibility intervals proposed in the NMP-Bench model. If some of them do not conform to the proposed ranges, it is recommended to cross-check the measuring instrument and the measuring process.

At this point, the NMP-Bench model for the musician *A* is complete. The results can be compared with respect to the sheet music, or with results achieved through other NMP tools for the same instrument and song. Obviously, the analysis can be supported by the use of graphs and tables to make the final understanding clearer and more fluent.

V. CASE STUDY

We explain our benchmark model and architecture by applying it to a concrete case study.

A. Test Case Description and Settings

We represent a scenario in which three musicians interact to play collaboratively. Musicians are simulated i.e., the musicians are actually three PCs which submit music tracks of individual instruments to the NMP tools. The first musician is an ASUS VivoBook Pr N56JN-CN048H in which a Windows 10 Home operating system is installed. The second is a WinBlu Energy L5 0111W7 where a Windows 7 Professional operating system is housed. Finally, the third is an ASUS F3M-AP034C-A with a Windows XP Professional operating system. These PCs are arranged and used in a single room. Regarding the connection, we rely on a Wi-Fi 802:11 b/g/n network with an average download speed of 8.99 Mbps and upload of 0.58 Mbps.

Three different music pieces are played, belonging to different genres. Specifically, these are: i) Stayin' Alive: disco song written and performed by the British group Bee Gees for the soundtrack of the movie Saturday Night Fever of 1977; ii) Sultans of Swing: a musical piece by the British rock group Dire Straits based on their self-titled 1978 debut album; iii) Superstition: song of American soul-funk songwriter Stevie Wonder from the 1972 Talking Book.

Each musician has one instrument to play when the music performance starts, for each song. Specifically we consider the following instruments: i) Stayin' Alive has bass, drums and guitar tracks, ii) Sultans of Swing has bass, drums and guitar tracks, and iii) Superstition instead has bass, drums and clavinet tracks. We assign these tracks to the three musicians respectively in the order in which they are enlisted. During the execution of the NMP music performance, the three musicians transmit to the NMP server their tracks, with the aim of interpreting the song in the best possible way. We consider this aspect a reasonable approximation of the individual interpretation of the music piece by musicians, given the absence of a team of professionals.

To collect data, we will use two software tools. The first tool is Wireshark [32], a software for packet sniffing and protocol analysis. Wireshark makes the computation of network performance metrics straightforward. It is important to observe that Wireshark is active only on the ASUS VivoBook Pr N56JNCN048H to record all network operations during the whole process. This notebook was selected because of its performance; on such notebook, Wireshark is low intrusive, requiring physical memory of a maximum of 77.8 MB and with a maximum disk usage of 0.1 MB/s[3] in our tests. The second tool helps us to detect the bpm related to a song: we use the BPM Counter of Abyss Media Company [12]. A tool as BPM Counter facilitates the computation of metrics for user satisfaction and musical scope.

The music performance will be executed using two different NMP tools, that we call TOOL A and TOOL B. We anonymize the names of the tools because the scope of this work is showing the NMP-Bench benchmark and its application, and not provide recommendations about tools. To have scientific relevance, such recommendations would require i) a lengthy discussion on the technical insights of the tool to motivate values collected, which is not doable within the page limit, and ii) above all, the involvement of a set of professional musicians, which is beyond the scope of this work.

Free network servers made available for the NMP tools are selected and used in our experiments. The three PCs are interacting with such servers for our experimental campaign.

B. *Results and discussion*

For simplicity, we indicate the three musicians with the letters *A*, *B* and *C* while the three tracks chosen for the test as *SA* (Stayin' Alive), *SoS* (Sultans of Swing) and *SU* (Superstition).

Network performance. Results are summarized in Table III. First of all, it is possible to observe that the value of data σ in Table I is here not available. This is a consequence of the type of codec that both NMP tools use. They implement lossy data compression algorithms i.e., part of the original information is lost during data compression and decompression. This aspect has an important influence on the quality of the compressed object and prevents a possible estimate of the data compression ratio.

TABLE III. NETWORK PERFORMANCE: VALUES COMPUTED FOR THE THREE MUSIC PIECES WITH THE TWO NMP TOOLS.

		TOOL A		TOOL B	
Attribute	**Metrics**	**Values**	**Data**	**Values**	**Data**
Responsiveness	$delay_{OOSE}$	~89.34 ms	N = 3, P = 1024 samples, R = 44.1 kHz, B = 2.5 blocks, C = 0.44 Mbps, Ch = 2, O = 384 bits, δ = 16 bits/sample, σ = n.d.	~193.86 ms	N = 3, P = 1024 samples, R = 44.1 kHz, B = 16 blocks, C = 0.17 Mbps, Ch = 4, O = 432 bits, δ = 16 bits/sample, σ = n.d.
	RTT	n.a.		[80, 86] ms	*RTT from Wireshark*
Efficiency	Channel utilization	[33, 34]%	br = [309, 338] kbps, L = [102, 112] byte	[84, 85]%	br = 146-161 kbps, L = 629-649 byte
Packet loss	Packet loss rate	n.d.	n.d.	[0.1, 0.3]%	Obtained directly from *Wireshark*
Productivity	Bandwidth	[0.29, 0.33] Mbps	CPS = [11, 15] Mbyte, RCT = [299, 369] s	[0.04, 0.05] Mbps	CPS = [1.6, 2.5] Mbyte, RCT = [328, 399] s
	Throughput	n.d.		[0.32, 0.34] Mbps	$RWIN$ = ~27 kbit

The values of RTT, packet loss rate and throughput for TOOL A are not available. Also in this case there is an explanation and this is to be found in the type of protocol used, which is UDP, that does not allow to collect these values. Finally, we computed $delay_{OOSE}$ following the indications reported in Section 3.1.1, and using as inputs the information on the system in our possession. The same approach was applied consistently to compute the $delay_{OOSE}$ in both tools. Noteworthy, a reliable computation of $delay_{OOSE}$ is somehow difficult without ad-hoc instruments; however, we believe our estimation is sufficient to explain how to use the benchmark.

With the only objective of showing that graphs can support tables and visualize the difference between tools as stated at the end of Section 4, we depict the measured bandwidth for TOOL A (black) and TOOL B (grey) in Figure 3.

All values retrieved are within the interval of plausible values of Table I, Table II.

Musical scope. From an analysis of tracks recorded during the music performance, it was possible to compute musical scope metrics (metronomic). Values are in Table IV. Due to the characteristics of our case study, that submits audio tracks to the NMP servers instead of having real musicians, the values of Table IV correspond for both TOOL A and TOOL B. This is the consequence of having used recorded tracks to simulate the musicians. These values are obtained with 18-seconds time window for each track.

Effectiveness. Effectiveness metrics to describe accuracy and time attributes have been recorded and computed, using the same time window used in the previous experiments described

TABLE IV. MUSICAL SCOPE FOR BOTH TOOLS, FOR THE THREE TRACKS CONSIDERED.

Track	Max. BPM	Average BPM	Min. BPM	BPM trend
SA	159 bpm	113 bpm	88 bpm	104 bpm
SoS	152 bpm	136 bpm	92 bpm	149 bpm
SU	160 bpm	107 bpm	94 bpm	100 bpm

TABLE V. EFFECTIVENESS (MEASURED FOR EACH INDIVIDUAL MUSICIAN) EXTRACTED FROM THE ANALYSIS OF THE AVAILABLE TRACKS.

Track	Musician	Regularity	Pacing (bpm)	Slope (bpm)
SA	A	0.79 %	110 bpm	3 bpm
	B	0.36 %	105 bpm	8 bpm
	C	0.91 %	127 bpm	-14 bpm
SoS	A	0.80 %	124 bpm	12 bpm
	B	0.38 %	148 bpm	-12 bpm
	C	0.71 %	136 bpm	0 bpm
SU	A	0.69 %	115 bpm	-8 bpm
	B	0.41 %	101 bpm	6 bpm
	C	0.44 %	107 bpm	1 bpm

TABLE VI. USER SATISFACTION FOR INTERACTIONS BETWEEN MUSICIANS EXTRACTED FROM THE ANALYSIS OF THE AVAILABLE TRACKS.

Track	Interaction	Asymmetry	Single imprecision
SA	AB	12 bpm	20 bpm
	BC	26 bpm	29 bpm
	AC	22 bpm	26 bpm
SoS	AB	24 bpm	34 bpm
	BC	14 bpm	26 bpm
	AC	30 bpm	39 bpm
SU	AB	16 bpm	27 bpm
	BC	6 bpm	11 bpm
	AC	17 bpm	26 bpm
Track	**Interaction**	**Ensemble Imprecision**	
SA	ABC	25 bpm	
SoS		33 bpm	
SU		21 bpm	

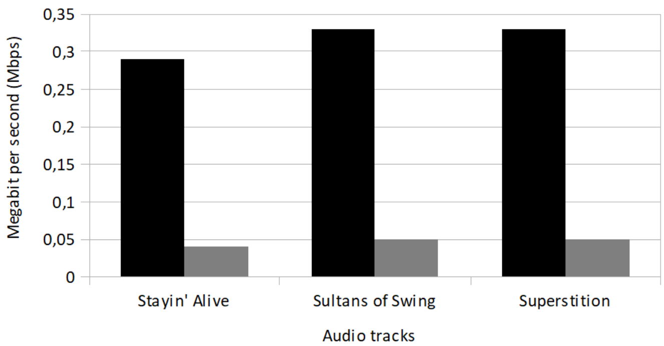

Figure 3. Comparison of bandwidth data. Results for TOOL A are represented in black, while results for TOOL B are in grey.

in this section. Values are the same for both tools, for the reason already explained.

Consequently, showing the difference between the studio track and the recorded track for Table V would simply lead to a set of 0s. For the sake of clarity, we show instead the value measured for each individual track in Table V. Table VI instead shows the difference between instruments (different tracks for each music song). As it can be easily identified from Table V and Table VI, the different tracks show high values for asymmetry and imprecision. This is not attributable to the performance of the NMP tools, but to the quality of the recorded tracks we used, for the reasons already explained.

C. Possible Consequences from the Application of the Benchmark

We briefly argument on possible actions and considerations from the application of the benchmark.

Our experiments show that results are strongly dependent from the quality of the connection: a bad connection may impact the quality of the music performance, and a general low satisfactory level may be due to the poor network of just one participant. As network performance is generally easy to measure, it is possible to check if benchmark results are biased or altered by slow network connection.

Additionally, the achieved measurements are useful to guide the possible development of the tools. For example, it is easy to understand what would be the optimal values in Table IV to Table VI, then evaluate which values are farther from the optimal, and finally research possible alternative solutions to be implemented in the tool. Also, comparison between tools facilitate understanding weakenesses and strengths of tools; a clear example is the difference in the bandwidth usage of the two tools, and also in the different values in Table III for the two tools.

VI. LIMITATIONS OF THE STUDY

To the best of our knowledge, NMP-Bench is the first study that proposes a benchmark of server-based NMP tools. It is then unavoidable that the study includes limitations, which are here attentively discussed, and that can be mitigated only through further research work. We distinguish between limitations of the model and limitations of our set-up and experimentation.

Considering limitations of the model, we first observe that our benchmark is focused on the performers. NMP-Bench model is intended to evaluate if the performer is satisfied, while there is limited or no attention to the experience of a potential listener. This is generally in line with the current application of NMP tools, that is still mostly for recreational initiatives of group bands, but it cannot be neglected in a long-term perspective, e.g., it has been also strongly pushed by the recent COVID-19 quarantine [37].

Second, a reliable way to compute the network performance metric d_{OOSE} should be included in the NMP-Bench architecture, possibly with the definition of specific tools or software procedures that are able to interact with hardware sound boards or drivers to collect the relevant time values.

Third, NMB-Bench requires an efficient method for computing the musical scope metrics, which is still an open problem in the scientific literature e.g., for what concern beat tracking. For example, the tool we used for beat tracking has some limitations, as for example it is possible that it return zero for audio files with variable tempo or with silent pauses (see the F.A.Q. at [12]).

Last, it is generally acknowledged that the success of any experimental data collection is strongly dependent on the quality of the setup, which should be carefully evaluated and validated e.g., through tests and checklist, before executing the experiments. Guidelines for quality assurance when using an NMP system would be desirable. While this constitute possible future works, it is currently not included in our work.

Considering instead limitations of our set-up and experimentation, we observe that the model is tested without involving actual musicians. This simulated setting reduces the relevance of the experimental campaign we performed. While this is a major limitation of the study, it can be overcome only with the availability of musicians. In addition, the three PCs were located in the same room, which reduces the distributedness of the system that we should expect in typical setttings. Last, we considered a small workload of three music pieces. Workloads are a relevant part of each benchmark, because they may have a direct impact on results: it is intuitive that a larger workload should be defined, that can be representative of the different demand of the group bands e.g., be inclusive of different song styles, or consider band of different dimensions.

VII. CONCLUSIONS

Networked Music Performance (NMP) tools allow establishing a live interaction between musicians located in different places, and have to overcome severe technological challenges to provide a satisfactory experience to the musicians. In this paper we presented the NMP-Bench (Network Music Performance Benchmark) model and architecture for benchmarking server-based NMP tools. To the best of our knowledge, NMP-Bench is the first attempt to create a benchmark for NMP tools that is comprehensive of the different characteristics of the domain, thus including network performance, musical scope, and effectiveness.

NMP-Bench has been proposed and applied to a case study. Despite the limitations of the case study, mostly related to the absence of real musicians to practise with the NMP tools, we believe it is sufficiently representative to show that NMP-Bench offers a credible approach for anyone interested in methods and metrics for assessing NMP solutions.

However, as it is also evident from the limitations that we reviewed in Section VI, this paper cannot be considered the point of arrival for the definition of a benchmark of NMP tools. Additional research contributions are required to present a mature benchmark that can be applied to server-based NMP tools, and in our opinion, especially two major research actions can be identified for the near future, that target the set-up and execution phase, and the data analysis phase. In fact, the first action is related to the support for a proper set-up and execution. As it is true for any measurement instrument [27], the correct set-up and the proper operation of the NMP tool and the NMP-Bench model are fundamental for the collection of valid results. For example, it is sufficient to consider the case of a slow Operating System that may introduce bias in the entire experimental campaign [22]. The operator could be supported with facilities (software utilities, checklists, and tests list) that minimize mistakes and allow validating the settings before the experiments are executed. The second action instead is related to the software support that our benchmark could offer for data collection and analysis. At present, the application of NMP-Bench relies mostly on the capacity of the operator to properly setup the entire model and run experiments. While clearly benchmarking requires experienced personnel, the support of software to perform measurements and show results would resolve many possible ambiguities and possible mistakes of the operator.

REFERENCES

[1] G. Weinberg, "The Aesthetics, History and Future Challenges of Interconnected Music Networks," *in Proceedings of the International Computer Music Conference* (2002).

[2] L. Turchet, "Smart Musical Instruments: Vision, Design Principles, and Future Directions," *IEEE Access*, vol. 7, pp. 8944-8963 (2018 Oct..). https://doi.org/10.1109/ACCESS.2018.2876891.

[3] D. Mills, " RFC-958 Network Time Protocol (NTP)," Internet Engineering Task Force (1985 Sept.).

[4] B. Wah, X. Su, D. Lin, "A Survey of Error-Concealment Schemes for Real-Time Audio and Video Transmissions over the Internet," *in Proceedings of the International Symposium on Multimedia Software Engineering*, pp 17-24 (Taipei, Taiwan), (2000 Dec.).

[5] C. Perkins, O. Hodson, V. Hardman, "A Survey of Packet Loss Recovery Techniques for Streaming Audio," *IEEE Network*, vol. 12, no. 5, pp. 40-48 (1998).

[6] Y. Chen, B. Chen, "Model-based Multirate Representation of Speech Signals and its Application to Recovery of Missing Speech Packets," *IEEE Trans. Speech Audio Process*, vol. 5, no. 3, pp. 220-231 (1997). https://doi.org/10.1109/MMSE.2000.897185.

[7] J. Kurose, K. Ross, *Computer Networking: A Top-Down Approach* (Pearson, 2012)

[8] C. Rottondi, C. Chafe, C. Allocchio, A. Sarti, "An Overview on Networked Music Performance Technologies," *IEEE Access*, vol. 4, pp 8823-8843 (2016 Dec.). https://doi.org/10.1109/ACCESS.2016.2628440

[9] A. Carôt, C. Werner, "Network Music Performance - Problems, Approaches and Perspectives," *in Proceedings of the Music in the Global Village*, vol. 162, pp 10-23 (Budapest, Hungary, 2002).

[10] L. Gabrielli, S. Squartini, "Wireless Networked Music Performance," *in Proceedings of the Wireless Networked Music Performance*, Springer (2016). https://doi.org/10.1007/978-981-10-0335-6_5.

[11] J. Padhye, V. Firoiu, D. Towsley, J. Kurose, "Modeling TCP Throughput: A Simple Model and its Empirical Validation," *in Proceedings of ACM SIGCOMM Computer Communication Review*, vol. 28, no. 4, pp 303-314 (1998 Oct). https://doi.org/10.1145/285237.285291.

[12] Abyss Media Company, "BPM Counter," https://www.abyssmedia.com/bpmcounter (accessed April 28, 2021).

[13] S. Cheung, Performance of the Stop-and-Wait Protocol, Course CS455: Introduction to Computer Network, Emory University (Atlanta, USA, 2016).

[14] M. Weik, Computer Science and Communications Dictionary (Springer Science & Business Media, 2001). https://doi.org/10.1007/1-4020-0613-6.

[15] M. Seltzer, et al. "The case for application-specific benchmarking." *In Proceedings of the Seventh Workshop on Hot Topics in Operating Systems* (1999 Mar.). https://doi.org/10.1109/HOTOS.1999.798385.

[16] V. Fischer, "Case Study: Performing Band Rehearsals on the Internet with Jamulus," http://llcon.sourceforge.net/PerformingBandRehearsalsontheInternetWithJamulus.pdf (accessed April 28, 2021).

[17] A. Carôt, C. Werner, "Distributed Network Music Workshop with Soundjack," *in Proceedings of 25th Tonmeistertagung* (Leipzig, Germany, 2008).

[18] Cockos Incorporated, "NINJAM: About," https://www.cockos.com/ninjam/ (accessed April 28, 2021).

[19] D. Akoumianakis et al., "The Musinet Project: Addressing the Challenges in Networked Music Performance Systems," *in Proceedings of the International Conference on Information, Intelligence, Systems and Applications*, pp. 1-6 (2015 July). https://doi.org/10.1109/IISA.2015.7388002.

[20] F. Meier, M. Fink, and U. Zólzer, "The JamBerry- Stand-Alone Device for Networked Music Performance based on the Raspberry Pi," *in Proceedings of the Linux Audio Conference* (Karlsruhe, Germany, 2014).

[21] V. Jacobson, R. Braden, D. Borman, "RFC-1323 TCP Extensions for High Performance", Internet Engineering Task Force (1992).

[22] L. Turchet, C. Fischione, "Elk Audio OS: an Open Source Operating System for the Internet of Musical Things," ACM Tran. on the Internet of Things (2021 Mar.). https://doi.org/10.1145/3446393.

[23] J. Lewis, "IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use", *Int. Journal of Human-Computer Interaction*, vol 7, no. 1, pp 57-78 (1995). https://doi.org/10.1080/10447319509526110.

[24] W. Wetzlinger, A. Auinger, M. Dörflinger "Comparing Effectiveness, Efficiency, Ease of Use, Usability and User Experience when using Tablets and Laptops," *in Proceedings of the International Conference of Design, User Experience, and Usability*, pp. 402-412 (2014). https://doi.org/10.1007/978-3-319-07668-3_39.

[25] S. Hart, "NASA-Task Load Index (NASA-TLX) - 20 Years Later", *in Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, pp. 904-908 (2006). https://doi.org/10.1177/154193120605000909.

[26] C. Bartlette, D. Headlam, M. Bocko, G. Velikic, "Effect of Network Latency on Interactive Musical Performance," *Music Perception: An Interdisciplinary Journal*, vol. 24, no. 1, pp 49-62 (2006). https://doi.org/10.1525/mp.2006.24.1.49.

[27] JCGM, International Vocabulary of Metrology - Basic and General Concepts and Associated Terms (VIM 3rd edition), (2012)

[28] S. Farner, A. Solvang, A. Saebø, P. Svensson, "Ensemble Hand-Clapping Experiments under the Influence of Delay and Various Acoustic Environments," *J. Audio Eng. Soc.*, vol. 57, no. 12, pp. 1028-1041 (2009 Dec.).

[29] P. Burk, "Jammin'on the Web-a new Client/Server Architecture for Multi-User Musical Performance," *in Proceedings of the International Computer Music Conference* (2000).

[30] A. Bondavalli, et al., "A new Approach and a Related Tool for Dependability Measurements on Distributed Systems," *IEEE Tran. Inst. and Meas*., vol. 59, no. 4, pp. 820-831 (2010 Apr.). https://doi.org/10.1109/TIM.2009.2023815.

[31] F. Brancati, A. Bondavalli, "Practical Aspects in Analyzing and Sharing the Results of Experimental Evaluation," *In Proceedings of the IEEE Symposium on Reliable Distributed Systems*, pp. 328-332 (2010 Oct.). https://doi.org/10.1109/10.1109/SRDS.2010.46

[32] U. Lamping, R. Sharpe, E. Warnicke, "Wireshark User's Guide for Wireshark 3.5," https://www.wireshark.org/docs/wsug_html_chunked/ (accessed April 28, 2021).

[33] D. Cameron, K. Potter, G. Wiggins, M. Pearce, "Perception of Rhythmic Similarity is Asymmetrical, and Is Influenced by Musical Training, Expressive Performance, and Musical Context," *Timing & Time Perception (T&TP)*, vol. 5, no. 3-4, pp. 211-227 (2017 Dec.). https://doi.org/10.1163/22134468-00002085.

[34] F. Cristian, C. Fetzer, "The Timed Asynchronous Distributed System Model," *IEEE Tran. Par. Dist. Systems*, vol. 10(6), pp. 642-657 (1999 Jun.). https://doi.org/10.1109/71.774912.

[35] C. Chafe, J. Caceres, M. Gurevich, "Effect of Temporal Separation on Synchronization in Rhythmic Performance," *Perception*, vol. 39(7), pp. 982-992 (2010 Jan.). https://doi.org/10.1068/p6465.

[36] L. Turchet, et al., "Internet of Musical Things: Vision and Challenges," *IEEE Access*, vol. 6, pp. 61994-62017 (2018 Sep.). https://doi.org/10.1109/ACCESS.2018.2872625.

[37] R. Wilson, "Aesthetic and Technical Strategies for Networked Music Performance," *AI & society*, pp. 1-14 (2020 Nov.). https://doi.org/10.1007/s00146-020-01099-4.

[38] M. Fink, U. Zölzer, "Low-Delay Error Concealment with Low Computational Overhead for Audio over IP Applications," *in Proceedings of the 17th International Conference on Digital Audio Effect*, pp. 309-316, (2014 Sep.).

[39] P. Verma, et al. "A Deep Learning Approach for Low-Latency Packet Loss Concealment of Audio Signals in Networked Music Performance Applications," *in Proceedings of the 27th Conference of Open Innovations Association*, pp. 268-275 (2020 Sep.). https://doi.org/10.23919/FRUCT49677.2020.9210988.

[40] ISO/IEC, ISO IEC 25010:2011 Systems and Software Engineering - Systems and Software Quality Requirements and Evaluation (SQuaRE) - System and Software Quality Models (2011).

BIOGRAPHIES

Matteo Gori

Matteo Gori is a former student **of** the University of Florence. Born on April 15, 1995 in Sansepolcro (Tuscany), in 2018 he graduated in Computer Science at the University of

Florence, with a Bachelor Thesis on Networked Music Performance. After graduating, he now works as an insurance consultant and business specialist at UnipolSai Assicurazioni SpA, an Italian company leader in the insurance sector.

Andrea Ceccarelli

Andrea Ceccarelli received the PhD in Informatics and Automation Engineering from the University of Florence, Florence, Italy, in 2012. He is currently a Research Associate of Computer Science at the same University. His primary research interests are in the design, monitoring and experimental evaluation of dependable and secure systems, and systems-of-systems. His scientific activities originated more than 100 papers which appeared in international conferences, workshops, and journals.