An overview on networked music performance technologies

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 4, 2016, accepted November 9, 2016, date of publication December 5, 2016, date of current version January 4, 2017.

Digital Object Identifier 10.1109/ACCESS.2016.2628440

An Overview on Networked Music Performance Technologies

CRISTINA ROTTONDI[1], CHRIS CHAFE[2], CLAUDIO ALLOCCHIO[3], AND AUGUSTO SARTI[4]

[1]Dalle Molle Institute for Artificial Intelligence, University of Lugano–University of Applied Sciences and Arts of Southern Switzerland, Lugano 6928, Switzerland
[2]Center for Computer Research in Music and Acoustics, Stanford. CA 94305-8180, USA
[3]Consortium GARR, Rome 00185, Italy
[4]Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milan 20133, Italy

Corresponding author: C. Rottondi (cristina.rottondi@supsi.ch)

ABSTRACT Networked music performance (NMP) is a potential game changer among Internet applications, as it aims at revolutionizing the traditional concept of musical interaction by enabling remote musicians to interact and perform together through a telecommunication network. Ensuring realistic performance conditions, however, constitutes a significant engineering challenge due to the extremely strict requirements in terms of network delay and audio quality, which are needed to maintain a stable tempo, a satisfying synchronicity between performers and, more generally, a high-quality interaction experience. In this paper, we offer a review of the psycho-perceptual studies conducted in the past decade, aimed at identifying latency tolerance thresholds for synchronous real-time musical performance. We also provide an overview of hardware/software enabling technologies for NMP, with a particular emphasis on system architecture paradigms, networking configurations, and applications to real use cases.

INDEX TERMS Music, audio systems, audio-visual systems, networked music performance, network latency.

I. INTRODUCTION

Networked Music Performance (NMP) represents a mediated interaction modality characterized by extremely strict requirements on network latency. Enabling musicians to perform together from different geographic locations requires capturing and transmitting audio streams through the Internet, which introduces packet delays and processing delays that can easily have an adverse impact on the synchronicity of the performance.

Computer-aided musical collaboration has been investigated starting from the early '70s, when musicians and composers, inspired by the electroacoustic music tradition, began exploiting computer technologies as enablers for innovative manipulations of acoustic phenomena (see [1] for a historical overview of related works of sonic art). In the past two decades the massive growth of the Internet has greatly widened the opportunities for new forms of online musical interactions. A categorization of computer systems for musical interactions is offered in [1], which include:

- local interconnected musical networks ensuring interplay between multiple musicians who simultaneously interact with virtual instruments [2];
- musical team-composing systems allowing for asynchronous exchange and editing of MIDI (Musical Instrument Digital Interface) data [3]–[5] or recreating virtual online rooms for remote recording sessions based on distributed systems connected through centralized servers[1];
- shared sonic environments, which take advantage of distributed networks by involving multiple players in improvisation experiments, as well as audience participation events [6]–[9];
- remote music performance systems supporting real-time synchronous musical interactions among geographically-displaced musicians.

NMP focuses on the last of the above categories and aims at reproducing realistic environmental conditions for a wide range of applications from tele-auditions, remote music teaching and rehearsals, to distributed jam sessions and concerts. However, several aspects of musical interactions must be taken into account. Musicians practicing in the same room rely on several modalities in addition to the sounds generated by their instruments, including sound reverberation within the physical environment and visual feedback from movements and gestures of other players [10]. Though communication technologies are still not sufficiently advanced to reliably and

[1]http://soundation.com

conveniently reproduce all the details of presence in musical performances, some technical necessities to enable remote interaction can be identified [11]. In particular, from the networking point of view, very strict requirements in terms of latency and jitter must be satisfied to keep the one-way end-to-end transmission delay below a few tens of milliseconds. According to several studies [12], [13], the delay tolerance threshold is estimated to be $20 - 30$ ms, corresponding to a distance of 8-9 m (considering the speed of sound propagation in air), which is traditionally considered as the maximum physical separation ensuring the maintenance of a common tempo without a conductor. However, this threshold varies a great deal depending on the abilities of the musician, his/her own stylistic expressions, and the strategies he/she applies to cope with delayed self-sound and other feedback affecting tempo. Several studies reported in [14] and [15] show that an asynchronism of up to $30 - 50$ ms (due either to the spatial dislocation of the performers, the delay in the auditory feedback introduced by the instrument itself, or the reaction delay elapsing between motor commands, the musician's corresponding haptic sensations, and audio feedbacks) are largely tolerated and even consciously emphasized to achieve specific stylistic effects. For example, organ players naturally compensate the delay between pressing on the keyboard keys and hearing the emitted sound, due to the great physical displacement between keyboard and pipes. The same applies to piano performance in which the time elapsed between the pressing of a key and the corresponding note onset varies between 30 and 100 ms according to the musical dynamics (sound loudness) and articulation (e.g. *legato, staccato*) [16].

In addition to subjecting players to remote topologies which violate their rhythmic comfort zones, a particularly hard digital challenge in NMP frameworks is synchronization of audio streams emitted by devices which do not share the same clock. Clock drift issues arise over minutes of performance which can create under-run (i.e. the condition occurring when the application buffer at the receiver side is fed with data at a lower bit-rate than that used by the application to read from the buffer, which obliges the application to pause the reading from time to time to let the buffer refill) or over-run conditions (i.e., when the buffer is fed at a higher bit-rate than the application reading rate, which leads to losses when incoming data find the buffer completely full). Operating systems of general purpose computers introduce processing delays up to a few milliseconds, which, in turn, affects the data acquisition and timestamping procedures at both capture and playback sides. In order to ensure an accurate stream alignment, signal processing techniques must be adopted to truncate or pad audio data blocks without impairing the perceptual audio quality by introducing artifacts. The same approaches must be adopted in case of network packet loss or to compensate the effects of network jitter: if a packet reaches its destination after its scheduled playback time, its audio data is no longer valid.

Given the wide variety of approaches to networked musical interactions which have been developed so far, the contribution of this survey is threefold. We first provide an in-depth analysis of the factors influencing musicians' latency acceptability thresholds. We then discuss the contributions to the overall delay experienced by NMP users along the audio streaming chain and identify system parameters affecting latency and perceived audio quality. We also provide a comprehensive overview of existing NMP frameworks and discuss hardware/software technologies supporting remote musical performances.

In particular, in Section II we discuss the methodologies for real-time audio streaming and strategies which minimize delay introduced by audio acquisition, processing and transmission. In Section III we summarize the results of several perceptual studies aimed at identifying the ranges of latency tolerance in different NMP scenarios, focusing on the impact of environmental and instrumental characteristics (e.g. acoustic reverberation and timbre), musical features (e.g. rhythmic complexity) and interpretative choices (e.g. respective relationship of leading parts, presence of a conductor). Then, in Section IV we discuss the instrument-to-ear delay that is perceived by a NMP performer due to the various processing and transmission stages required for audio streaming, and identify the system parameters affecting such contributions. Several state-of-the-art NMP frameworks are comparatively reviewed in Section V, detailing their architectural, hardware and software characteristics. A discussion on future research directions in the field of NMP is provided in Section VI. Finally, conclusions are drawn in Section VII.

II. APPROACHES TO NMP: STREAMING, COPING, CONDUCTING, AND PREDICTION STRATEGIES

A. MUSICAL DATA STREAMING AND PREDICTION

Though the majority of NMP systems are designed to convert sound waves generated by a generic audio source into transmitted digital signals (which ensure full compatibility with non-electronic instruments and voice), alternative paradigms have also been considered to avoid transmission of audio streams through the network, thus reducing bandwidth requirements and improving scalability. Some frameworks use MIDI to transport synthetic audio contents, thus being suitable only for electronic instruments. Computer-controlled instruments such as *Yamaha Disklavier*[2] offer practical NMP capabilities: these pianos are equipped with a measurement unit storing the information derived from key shutters (usually located below the key and at the hammer shank), and a reproduction unit controlling solenoids below the back of each key. The gestural data measured during the pianist's performance can be communicated to a remote instrument via an Internet connection using a proprietary protocol, so that the key strokes actuated by the pianist are reproduced at the remote side.

A gestural data codification approach (using audio signal recognition) has been implemented for percussion [17] and is combined with a prediction mechanism based on the

[2]http://usa.yamaha.com/products/musical-instruments/keyboards/disklaviers/

analysis of previous musical phrases. Methodologies drawn from studies on computer accompaniment systems have been applied to NMP by modeling each performer as a local agent and recreating the performed audio at the remote side by means of a combination of note onset detection and score following techniques, based on pre-recorded audio tracks [18]. Bayesian networks for the estimation and prediction of musical timings in NMP have also been investigated [19].

Motion-tracking technologies have been employed in NMP to create graphics for remote orchestra conducting which outperform the latencies of traditional video acquisition [20], or for prediction of percussion hits based on a drummer's gesture, so that the sound can be synthesized at a receiver's location at approximately the same moment the hit occurs at the sender's location [21]. Frameworks for networked score communication/editing [22] and score-following [23] in support of NMP have also been developed.

B. STRATEGIES FOR DELAY COMPENSATION

In [24], according to the network latency conditions, playing strategies between pairs of musicians are categorized as follows: The first two of these result from acoustical situations with varying amount of delay between the players (temporal separation) and depend on the type of music (style, tempo, etc.). The last is a delay-compensating technique in which the acoustical situation is manipulated electronically to add a certain amount of self-delay to offset the perceived temporal separation of the other musician.

1) *realistic interaction*: this is the conventional musical interaction approach, enabling full, natural interplay between the musicians and implying no awareness of delay. Interactions of this type are the only way to achieve truly acceptable performances by professional players.
2) *master-slave*: One of a pair of players assumes a leading role and establishes the musical timing exclusively while ignoring the audio from the other player, who adapts to it and follows without significant inconvenience. Network delays can be tolerated up to a self-delay tolerance threshold (usually around $100 - 200$ ms [25]).
3) *delayed feedback*: an alternative approach consists in artificially adding a self-delay to the musicians' own audio feedback, up to an amount equal to the audio round trip time of the full NMP circuit. At that level, it is perceived as synchronized with audio generated by the remote counterpart. A variant to this solution adds a self-delay equal to the one-way network delay and requires the usage of a metronome (or any other cue) which must sound in perfect sync at both sides. However, such conditions may be difficult to achieve, due to drift issues in the synchronization of the local clocks [26].

III. UNDERSTANDING THE IMPACT OF LATENCY

Various studies of the effects of delay on live musical interactions have appeared in the last decade. Table 1 summarizes the characteristics of the tests reported in each of them, including the types of instruments and the musical pieces performed, the latency and tempo ranges tested, and the quality metrics applied for the numerical assessments. A first group of experiments focuses on rhythmic patterns performed by hand-clapping by both musicians and untrained subjects, [27]–[32], whereas other studies [25], [33], [34], [37], [39]–[41] consider a wide range of acoustic and electronic instruments, performing both predefined rhythmic sequences or classical/modern musical pieces. One work [35] focuses on theatrical opera pieces involving singers, a piano player and a conductor, and investigates the effects of network latency on both audio and video data. Combined transmission of audio and video data is considered in [37] and [41].

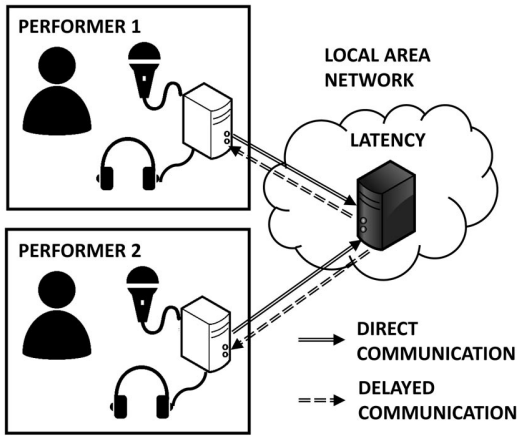

FIGURE 1. Typical testbed configuration for tester pairs. Latency can be introduced either by digital delay on the central experiment computer or by a network emulator (shown).

A. TEST SETUP AND DESCRIPTION

With the exception of [35], all the studies listed in Table 1 use similar test setups consisting of two sound-isolated anechoic rooms, each one hosting one musician, as depicted in Fig. 1. The subjects hear each other by means of headphones. Visual interactions between the musicians are prevented. Audio signals are captured by means of microphones, and either connected to a central experiment computer which inserts digital delay, or are directly converted in the room, packetized and transmitted via a wired Local Area Network (LAN) to the counterpart. Each subject hears his/her own instrument without additional delay, whereas the audio feedback from the counterpart is delayed by a delay which is electronically added by the central experiment computer or at the network interface via dedicated network impairment software (e.g. Netem [42]). Apart from the experiments in [39], where the behavior of a telecommunication network in terms of variable transmission delay and jitter is emulated by generating random packet delays according to a normal statistical distribution, the testbeds introduce constant packet delays.

TABLE 1. Summary of latency tolerance tests.

Authors	Hand Clapping	Instruments	rhythmic Patterns	BPM Range	One-Way Delay [ms]	Asymmetric Delay	Musical Pieces	Musical Features	Quality Metrics
Chafe, Gurevich *et al.* [27]–[30]	✓	✗	✓	60 – 120	1 – 77	✓	✗	✗	BPM trend, BPM slope
Farner *et al.* [31]	✓	✗	✓	86 – 94	6 – 67	✗	✗	Reverberation, subjects' musical training	BPM trend, BPM slope, initial BPM value, imprecision, asymmetry, subjective rating
Driessen *et al.* [32]	✓	✗	✓	90	30 – 90	✗	✗	✗	steady-state BPM, time difference, subjective rating
Bartlette *et al.* [33]	✗	violin, cello, flute, clarinet	✗	NA	6 – 206	✗	"Twelve Duets k.487", No.2 and No. 5 (W.A. Mozart)	Prior practice	Mean Pacing, Mean Regularity, Mean Asymmetry, Subjective rating
Carot *et al.* [34]	✗	percussions, bass, saxophone	✓	60 – 160	1 – 75	✗	"The days of wine and roses" (H. Mancini, J. Mercer)	reverberation, self-delay	Subjective rating
Barbosa *et al.* [25]	✗	violin, cello	✓	80	25 – 120	✗	✗	attack time	BPM trend, BPM dynamic time warping analysis
Olmos *et al.* [35]	✗	piano, singers, conductor	✗	NA	15 – 135 (audio), 60 – 180 (video)	✓	"Il core vi dono..", (W.A. Mozart, from "Cosi fan tutte"); "Ah! Voi signor" (G. Verdi, from "La Traviata"); "Bess you are my woman" (G. Gershwin, from "Porgy and Bess")	✗	Subjective rating, galvanic skin response, skin conductance response, BPM dynamic time warping analysis, BPM curvature points
Chew *et al.* [36]–[38]	✗	piano	✗	46 – 160	0 – 150	✗	Sonata for Piano Four-Hands (F. Poulenc)	prior practice, self-delay	Subjective rating, BPM segmental analysis
Rottondi *et al.* [39]	✗	piano, guitar, clarinet, violin, percussions	✗	80 – 132	15 – 75	✗	"Bolero" (M. Ravel), "Master blaster" (S. Wonder), "Yellow submarine" (The Beatles)	rhythmic complexity, spectrum statistical moments, musical part	BPM trend, BPM slope, subjective rating
Kobayashi *et al.* [40]	✗	MIDI piano	✗	N.A.	0 – 100	✗	demonstrative monodic music	✗	onset global and local phase difference

In [34] and [37], the addition of self-delay is introduced. In [31] and [34], configurations with artificially added reverberation are tested. The tests in [37] and [41] assume a combined streaming of unsynchronized audio and video data, but no specific investigation on the impact of video usage is provided. Conversely, the experiments in [35] require a testbed with three acoustically-insulated rooms, each one equipped with microphones/headsets, cameras and screens (see Fig. 2). Both synchronized and unsynchronized audio/video transmissions are tested. Each session involves 2-3 singers, a conductor and a pianist: the singers are located in two rooms and the conductor in the third room. According to the specific test session, the pianist performs either in the conductor's room or in one of the singers' rooms.

Almost all the reviewed studies report that the rhythmic patterns/music scores were provided to the testers in advance and that they were free to practice together in the same room until they felt comfortable with the performance. In [33] and [37], scenarios in which the testers could practice at each network latency level and develop strategies to compensate the audio latency are also considered. All the tests requiring the execution of rhythmic patterns consider the

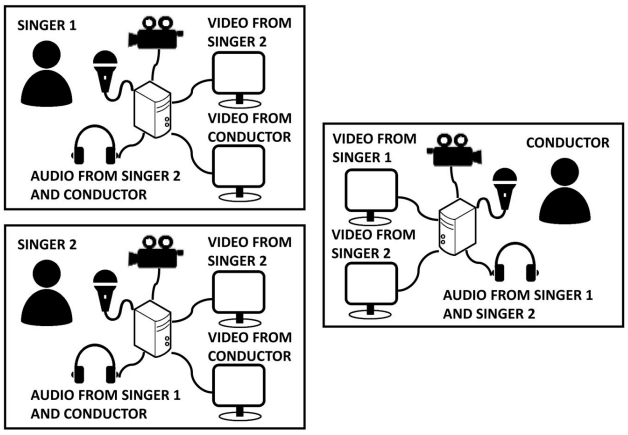

FIGURE 2. Testbed configuration for experiments in [35].

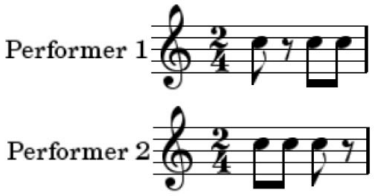

FIGURE 3. Rhythmic pattern considered in [27]–[32] and [34].

rhythmic structure reported in Fig. 3, whereas the repertoire used for the tests on musical pieces is reported in Table 1. Tests focusing on hand clapping consider a population of subjects with and without musical education, whereas the remaining ones assume prior musical education at amateur or professional level.

In all the experiments, the order of the tested network delay scenarios was randomly chosen and kept undisclosed to the testers, in order to avoid biases or conditioning. In [25], [27]–[32], [34], and [39], some bars of beats at the reference tempo δ expressed in Beat Per Minute (BPM) were provided to the players before the beginning of each single performance, either by an instructor or by means of a metronome. In [31]–[35], [37], and [39], once each trial was concluded, the subjects were asked to provide a subjective rating of the performance within a predefined scale of ranges or to answer to a qualitative questionnaire.

B. ANALYTICAL MODELS

Some insight into the impact of delay can be gathered in an analytical fashion by developing interaction models and studying their behavior. Two types of approaches have been used, one which uses signal-based, non-parametric methodologies based on measured time-series information [43]–[45] and the other starting from *parametric* representations of the nonlinear interaction between dynamical systems [32]. The latter approach derives a qualitative description of the impact of delay from a rough analytic prediction of the tempo evolution as a function of the network delay. Measured results are loosely in agreement.

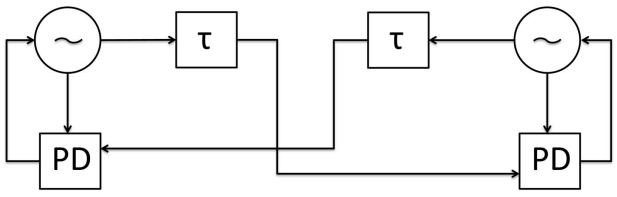

FIGURE 4. Coupled oscillators with delay τ controlled by phase detectors (PDs) [32].

In both [32] and [45], interacting performers are modeled by coupled oscillators as in Fig. 4. As we can see, each oscillator monitors its own pace as well as that of the other oscillator through a delayed observation. The frequency correction (control input) of each oscillator depends on the phase difference measured at the corresponding Phase Detector (PD) [32], [46], [47]. Each oscillator, however, has its own free running frequency, which is what is attained in the absence of input control. A closed-form analysis of the behavior of these coupled dynamical systems in [32] reveals that the oscillation frequency of the resulting system settles to the value

$$\Omega = \frac{\overline{\omega}}{1 + K\tau}, \tag{1}$$

where $\overline{\omega}$ is the mean value between the two natural oscillation frequencies, τ is the network delay and K is a constant that describes the state-update equations of the oscillators [32]. Accompanying experimental results obtained by directly measuring the steady-state tempo confirm the model's validity, though in other, and more complex tempo evaluation experiments [28], [39], [45], other phenomena seem to have an influence on the evolving tempo, possibly related to the role of the musician, the rhythmic complexity, and etc. This is further described in Section III-D.

An alternate approach to studying the impact of delay on the resulting tempo is described in [27]. Gurevich *et al.* modeled performers as memoryless systems that have no prior knowledge of the tempo and react instantaneously to the last detected beat without introducing arbitrary tempo deviations. Under such strong assumptions, the instantaneous tempo $\overline{\delta}$ of the musical performance can be computed as:

$$\overline{\delta}(n) = \frac{60}{60/\delta + n\tau} \tag{2}$$

where $60/\delta$ is the quarter note interval (in seconds) with reference tempo δ (in BPM), n is the number of elapsed quarter pulses and τ is the end-to-end delay, which is assumed to be two-way symmetrical. If $\tau = 0$, then $\overline{\delta}(n) = \delta$, otherwise $\overline{\delta}(n)$ decreases (less than linearly) with n. As performers tend to perceive tempo over intervals that are longer than a single beat, we expect them to largely outperform the model described by Equation 2. As a matter of fact, as confirmed by the numerical results shown in [27], the value of $\overline{\delta}(n)$ can be taken as a lower bound of the real performance tempo. In [45], the model was contrasted to a coupled oscillator model which includes anticipation. Predicted tempo values were closer (but not the same) as measured tempi.

C. QUANTITATIVE PERFORMANCE ASSESSMENT METRICS AND INDICATORS

The metrics proposed in the scientific literature to evaluate the quality of a networked musical interaction can be organized in two macro-categories, i.e. subjective and objective metrics. The former category includes opinion scores provided by the musicians to evaluate various aspects of their performance e.g., their emotional connection with the remote musician [35], the perceived delay [39] and level of asynchrony with respect to their counterpart [32], their willingness to use an NMP system based on their personal experience during the tests [32], or a global rating on the overall quality of experience [31], [33]–[35], [37], [39].

The latter category comprises numerical attributes extracted from the recorded audio tracks with the following

procedure: first, the time instants t_n in which the n-th quarter onset/hand-clap occurs are identified (either manually or by means of a peak search algorithm). Then, Inter-Onsets Intervals (IOIs, measured in seconds) between quarter notes are computed as $IOI_n = t_{n+1} - t_n$. Finally, the conversion of IOI to actual tempo (in BPM) is obtained as $\overline{\delta}(n) = 60/IOI_n$.

Based on $\overline{\delta}(n)$ and on the sequence of IOIs, the following metrics can be computed:

- *Pacing,* π [33]: mean IOI computed over the whole trace as $\pi = \frac{1}{N}\sum_{n=1}^{N} IOI_n$;
- *Regularity,* ρ [33]: coefficient of variability of the sequence of IOIs calculated as the IOI standard deviation to mean ratio.
- *Asymmetry,* α [28], [31], [33]: mean time that performer B lags behind performer A, measured as $\alpha = \frac{1}{N}\sum_{n=1}^{N}(t_n^B - t_n^A)$
- *Imprecision,* μ [31]: standard deviation of the inter-subject time differences, measured as $\mu = \sqrt{\frac{1}{N-1}\sum_{n=1}^{N}(t_n^B - t_n^A)^2}$
- *Tempo Slope,* κ [27], [29], [39]: the sequence $\overline{\delta}(n)$ can be linearly interpolated to estimate the tempo slope. Positive slopes indicate a tendency to acceleration, whereas negative slopes indicate a tempo decrease. Though the definition of tempo slope may at first glance appear contradictory w.r.t. the assumption of existence of an asymptotic steady-state tempo postulated in [32], it is worth noticing that in most cases the audio trace is divided in time windows of a few seconds duration and the trend of the actual tempo $\overline{\delta}(m)$ maintained during the m-th time window is obtained as a function of the average IOI over the onsets occurred within the window duration [25], [35]. It follows that the tempo slope may exhibit fluctuations over time. In particular, indications from experiments (e.g. the ones reported in [39]) show that negative slopes often occur during the first few seconds of networked musical interaction, especially in the presence of high latency. With the passing of time, the performance often tends to stabilize around a lower BPM, meaning that the slope κ assumes near-to-zero values, in accordance to the trend obtainable by means of equation (2). However, this is not always the case: in some conditions the players are unable to reach a stable tempo asset and the tempo trend exhibits a monotonically decreasing trend, which eventually leads to the interruption of the performance. Therefore, the steady-state tempo (when achieved) can be defined as the value of $\overline{\delta}(m)$ corresponding to near-to-zero values of the tempo slope κ.

Additional metrics (not discussed here) can also be extracted from the time warping analysis of the tempo trend $\overline{\delta}(n)$.

D. QUANTITATIVE AND QUALITATIVE RESULTS

The main goal of all the experiments is the identification of latency ranges allowing for satisfactory real-time musical interactions and the investigation of the impact on such ranges of musical features including the rhythmic complexity of the performed pieces, the effect of the instruments' timbral characterization and attack time, and the musical role of the performed part. Other factors affecting sensitivity to delay are the acoustical conditions (i.e. presence/absence of reverberation), the level of musical training of the performers and the role of delay-compensating strategies. In the following, we summarize the main outcomes of the surveyed experiments.

1) EFFECTS OF LATENCY ON THE PERFORMED TEMPO

Studies [27]–[31], [39] report the trend of the tempo slope κ averaged over multiple trials with reference tempo δ in the range $86 - 94$ BPM [27]–[31] and $60 - 132$ BPM [39].

In all the reported results, positive tempo slopes occur for latency values below $10 - 15$ ms. The authors of [28] postulate such behavior as the consequence of an intrinsic tendency to anticipate which has been already identified in studies on negative mean asynchrony in metronome-based tapping experiments (see [43] for an overview). In the range from $10 - 15$ to $20 - 25$ ms, performers are generally able to maintain a stable tempo, very close to the reference δ. Opinion ratings agree, providing positive/very positive evaluation ratings of the overall performance quality [31], [33], [39] and latency is either not perceived at all or is slightly noticed [39]. Delay becomes clearly perceivable in the range between $20 - 25$ and $50 - 60$ ms, when the quality of the performance starts deteriorating: the performers exhibit a pronounced tendency to decelerate (i.e., κ assumes negative values, whereas the pacing π increases) and their quality ratings consistently diminish. Moreover, the values of imprecision and asymmetry, which remained almost constant for delays below $25 - 30$ ms [28], [31], [33], [40], start rising. With delays above 60 ms, the performance is heavily impaired and the latency conditions are generally judged as barely tolerable [34], [37], [39]. Timings lose regularity even within a single part (i.e., ρ exhibit a remarkable increase above $60 - 80$ ms). Interestingly, the segmental tempo analysis conducted in [37] shows that the highest tempo variations occur in the case of delays in the range $50 - 100$ ms, whereas delays above 100 ms exhibit lower tempo variability, though the absolute tempo reduction is more consistent. As suggested by the authors, a possible explanation for this phenomenon is that such latency values are so unacceptable that the players were performing on ''auto-pilot'', disregarding the auditory feedback from the counterpart and only focusing on maintaining a stable tempo. This kind of behaviour emerges also from the analysis in [40], which shows that the average onset phase difference exhibits a peak in the range $50 - 80$ ms which then decreases again between $80 - 100$ ms, whereas its standard deviation remains almost constant in case of delays below 80 ms and rises sharply when latency exceeds 80 ms. The authors motivate these surprising results as follows: above 80 ms the synchronism of the performance is so compromised that phase differences between onsets start fluctuating, i.e. they may assume either positive or negative values (meaning that one performer may either be lagging or anticipating

the other, without a clear trend, thus explaining the high standard deviation), which brings the average onset phase difference closer to 0. These results seem to indicate that in the experiments conducted in [40] the switch to ''auto-pilot'' performing modality did not take place.

Some studies also evaluate the effect of asymmetric delays [29], [35], concluding that the effects are dominated by the impact of the highest network delay contribution among the two directions (forward/backward transmission).

2) IMPACT OF INSTRUMENT TIMBRE AND ATTACK TIME

Paper [33] makes a first attempt to investigate the impact of the instrument choice on the performance quality by comparing two executions of the same piece, performed by a string duo and a clarinet duo, respectively: strings exhibit higher deceleration and asynchrony compared to clarinets for all the tested latency conditions, though the subjective rating of the players is unexpectedly slightly higher for the strings duo. A more thorough investigation on the dependencies between performance quality and timbral attributes of the played instruments is proposed in [39], where the timbre of a set of seven instruments is characterized by means of the first four statistical moments of their spectrum magnitude. The study shows that instruments with high spectrum entropy and flatness (which are widely used as indicators of sound noisiness) are more prone to tempo deceleration for latencies above 35 ms. The corresponding quality ratings provided by the performers also exhibit a decrease when spectrum entropy and flatness increase. The same considerations hold for instruments with high values of the spectral centroid, which is related to the sound brightness.

Paper [25] specifically investigates the correlation between instruments' perceptual attack times and sensitivity to network delay: experiments are conducted by asking two players to modulate the attack strokes of a violin and a cello when performing with a bow. Results show that, for a given latency value, slow attack times lead to a more pronounced deceleration w.r.t. sharp attack times. However, a better synchrony is achieved in case of slow attack times. Though the above mentioned experiments have highlighted that attack times have an impact on the performance quality, a more in-depth analysis is required to verify the applicability of the results to a wider range of musical instruments. For example, the usage of electronic instruments with settable/tunable attack times would make them independent of the execution style of the single performers, thus enabling a more objective evaluation of the effects of the attack time variation.

3) IMPACT OF REVERBERATION EFFECTS

Though all the tests described take place in semi-anechoic or anechoic rooms, two studies compare the results with those obtained by adding artificial reverberation: paper [34] reports that no noticeable improvements were observed by the players when performing with reverb, [31] finds an increase in asymmetry and a decrease in the regularity within the single parts in anechoic conditions, whereas artificial reverberations caused a slight decrease in the initial tempo (evaluated over the first 5 onsets of each performance).

4) IMPACT OF RHYTHMIC COMPLEXITY

In [29], [31], [34], and [39], experiments are conducted where, for each latency value, multiple executions of the same rhythmic structure or musical piece with different reference tempi are performed. Paper [29] shows that, when performing a fixed rhythmic pattern, increasing the reference tempo while maintaining a constant network latency leads to a decrease of the slope κ. Similar results are provided in [39], where the performed musical pieces are rhythmically characterized by means of the mean event density (i.e., the number of distinct onsets per second) and rhythmic complexity (which is a function of the reference BPM δ and of the rhythmic figures appearing in the score). Therefore, the latency tolerance thresholds decrease when the reference BPM increases, as shown by the performance quality ratings provided by the musicians in [34] and [39]: as an example, drummers/bassists performing a succession of quarter notes at $\delta = 60$ BPM could on average tolerate latencies up to 40 ms, which reduced to only 15 ms (on average) when the reference tempo was doubled to $\delta = 120$ BPM.

Moreover, as reported in [31] and [39], in the presence of network delay the higher is the reference BPM value, the lower is the initial tempo at which the musicians performed the first few measures (typically corresponding to the first $5 - 10$ s of musical interaction). These results indicate that the influence of latency is immediately perceived by the performers, who start adjusting their tempo from the very first measure. Close examination of the first cycle of beats in [45] reveals a switch in phase adaptation (anticipation) with different delay conditions that is almost instantaneous.

5) IMPACT OF MUSICAL TRAINING, PRIOR PRACTICE, AND LATENCY COMPENSATION STRATEGIES

The hand-clapping experiments discussed in [31] included two sets of subjects, grouped based on their musical education level. Results show that musicians are more sensitive to latency, since on average their performance exhibits a more pronounced deceleration with respect to non-musicians for a given delay value, and for non-musicians the average asynchrony is higher.

The effect of prior training with various network delay configurations is investigated in [33] and [37]: the first study shows that allowing the players to practice at each latency level, possibly developing common strategies to cope with the delay, did not lead to noticeable difference in the delay tolerance thresholds, whereas the second reports that prior practice reduces the tempo deceleration, improves the synchrony among the two players and the regularity within a single part (with more pronounced improvements for strings as opposed to clarinets), whereas no significant variation of the quality rating is registered. It is worth noting that all the considered works address the three delay compensation strategies enumerated in Section I, whereas [34] and [37]

consider the introduction of an additive self-delay at one of the two sides. Such strategy leads to a considerable increase of the latency acceptance thresholds and quality rating (up to 65 ms in [37], where the players claim that the performance could have become ''perfect'' with further practice, and up to 190 ms in [34], though the testers defined the scenarios where self-delays exceeded 30 ms as ''unnatural''). Conversely, the adoption of self-delay at both sides led to much lower latency tolerance levels, (at most 80 ms, as reported in [34]) and such configuration was considered unacceptable by a remarkable portion of the tested subjects. Such results confirm the ones obtained in [25] by imposing a delayed auditory self-feedback during solo performances (i.e., the sound produced by the performer's own instrument is delayed by a fixed time lag): the tolerance ranges vary from 60 to 200 ms, depending on the reference tempo and type of instrument.

Though some experiments dedicated to the evaluation of the dependency of steady-state tempo and self-delay in a solo performance have appeared in [25], none of the above mentioned studies attempted to identify a correlation between the maximum individual latency tolerance applied to the auditory feedback from the musicians' own instrument and the quality of their performance when interacting with a counterpart in the presence of latency. Since the self-delay tolerance is highly subjective and depends on a variety of factors including the instrument type and the proficiency of the musician, it could indeed serve as benchmark to remove the biases introduced by such factors during a networked musical interaction.

Finally, the presence of a conductor avoids the recursive drag on tempo by providing a common cue to the performers: results reported in [35] counterintuitively show a tempo increase for high network delays compared to a benchmark condition in which no network delay is imposed. This scenario leads to a variation of the master-slave strategy where the master role is taken by the conductor him/herself. However, in order to maintain a stable tempo, the conductor typically ignores the audio feedback from the performers, therefore she/he cannot adopt any correction strategy as a reaction to the performers' execution but relies exclusively on his/her inner feel.

6) IMPACT OF COMBINING AUDIO-VIDEO DATA

The only investigation found to date regarding the impact of the de-synchronization of audio and video data is [35], which focuses on opera performances. The singers, conductor and pianist receive both audio and video feedback from two remote locations: audio and video can be either synchronized or manipulated for different latency values (within ranges of $15 - 135$ ms for audio and of $60 - 180$ ms for video). Though the performers claimed that they attributed more importance to visual contact than to the audio feedback and that they generally referred to one single cue source (chosen among the conductor's gesture, the piano accompaniment or the audio/gesture of the singing partner) while ignoring the others, the results of the set of experiments did not lead to a clear identification of a preferred combination of modality types and manipulated delays e.g., syncronized audio but delayed video, syncronized audio-video but with higher delay, etc. Measurements of the electrodermal activity (i.e. the continuous variations in the electrical characteristics of the skin) of the performers through galvanic and conductance skin responses, which provide an indication of the degree of excitation of a subject, reported a higher level of stress when the pianist was not located in the same room of the conductor, possibly leading to unsyncrhonization between the musical accompaniment and conductor's cueing gestures. A comprehensive assessment of the importance of video signals and the effects of audio-video misalignment remains to be investigated. It is still unclear under what conditions combining audio and video data will improve or negatively affect networked music interactions.

IV. CONTRIBUTIONS TO END-TO-END DELAY

In NMP, the overall delay experienced by the players includes multiple contributes due to the different stages of the audio signal transmission: the first is the delay introduced by the audio acquisition/playout, processing, and packetization/depacketization at sender/receiver sides; the second is the pure propagation delay over the physical transmission medium; the third is the data processing delay introduced by the intermediate network nodes traversed by the audio data packets along their path from source to destination, the fourth is the playout buffering which might be required to compensate the effects of jitter in order to provide sufficiently low packet losses to ensure a target audio quality level.

More specifically, as depicted in Figure 5 the Over-all One-way Source-to-Ear (OOSE) delay perceived by the users of an NMP framework includes:

1) in-air sound propagation from source to microphone;
2) transduction from the acoustic wave to electric signal in the microphone (negligible);
3) signal transmission through the microphone's connector (negligible);
4) analog to digital conversion (possibly with encoding) and internal data buffering of the sender's sound card driver;
5) processing time of the sender's machine to packetize the audio data prior to transmission;
6) network propagation, transmission and routing delay;
7) processing time of the receiver's PC to depacketize (and possibly decode) the received audio data;
8) application buffering delay;
9) driver buffering of the receiver's sound card and digital to analog conversion;
10) transmission of the signal through the headphone or audio monitor (loudspeaker) connector (negligible);
11) transduction from electric signal to acoustic wave in the headphones or loudspeaker (negligible);
12) for loudspeakers, in-air sound propagation from loudspeaker to ear.

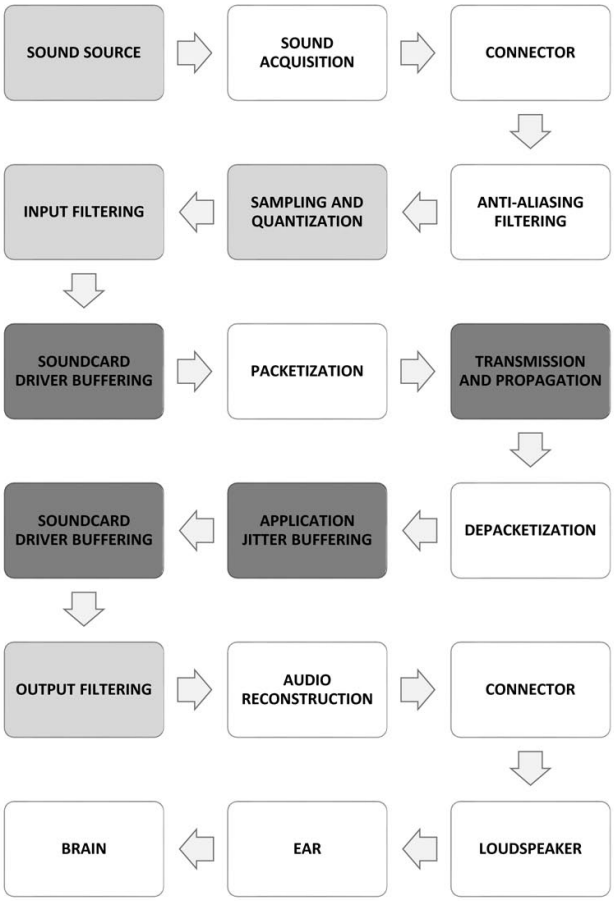

FIGURE 5. **Contributions to OOSE delay [12]. Darker background colors indicate higher delay contributions.**

TABLE 2. **List of symbols.**

Symbol	Description	Range of Values
R	soundcard sampling rate	$8-96$ KHz
L	soundcard sampling resolution	$16-24$ bits/sample
F	soundcard I/O filter length	$50-100$ audio samples
P	soundcard block size	$64-512$ samples
η	codec compression ratio	$0.25-1$
Ch	number of audio channels	$1-16$
B	application buffer size	$2-16$ blocks
H	total packet data overhead	<1000 bits
C	bandwidth of the network interface card	$0.054-20$ Mbit/s

Most of the above listed contributions depend on system parameters such as the audio card sampling rate and resolution, the audio block and buffer sizes (see Table 2), whereas others (e.g. the network routing delay) are independent of the system design and cannot be directly controlled. By tuning such parameters, the experienced end-to-end delay may vary significantly. However, latency savings can usually be achieved only at the expense of the audio quality level. Therefore, a trade-off emerges between system latency, bandwidth requirements and audio quality. In the following, we discuss in detail the contributions of each stage to the total delay budget and their dependencies on the NMP system parameters.

A. AUDIO ACQUISITION AND DIGITIZATION

By varying the distance between the audio source and the microphone, and the loudspeaker and the user's ears on the receiving side, the in-air sound propagation delay can be made arbitrarily low with an intelligent placement of the transducers. Therefore, in the following we will assume that the in-air propagation delay prior to the audio signal acquisition is negligible.

Typically, the soundcard first applies an analog anti-aliasing low pass filter [48], then samples the signal at rate R, and quantizes each sample using a given number of bits, L (i.e., using 2^L quantization levels). The delay D_a introduced by these stages is given by the sampling time $1/R$ (e.g., 20.83 μs for $R = 48$ kHz). An optional coding stage can be introduced here: audio codecs implement algorithms to (de)compress audio data according to a given coding format with the objective of representing the signal with the minimum number of bits while preserving high-fidelity quality, in order to reduce the bandwidth required for transmission. Audio codecs usually rely on sub-band coding, which enables for inclusion of pyscho-perceptual models. The more the sub-bands, the higher is the achievable compression ratio, η.[3] However, increasing the number of sub-bands also leads to higher encoding/decoding algorithmic delays, D_c: a comparison of the performance of the most widely used high-fidelity formats (e.g. MPEG/MP3 [49]) reported in [50] shows intrinsic latencies of at least 20 ms, which is hardly acceptable for the OOSE delay budget. Therefore, low-delay codecs specifically addressing latency intolerant applications have recently been developed: the OPUS [51], [52] (evolved from the prior CELT [53]), ULD [54] and Wavpack [55] codecs achieve algorithmic delays as low as $4-8$ ms. A thorough performance assessment of the packet loss concealment techniques implemented in the OPUS codec is reported in [56]. Modifications of the ULD codec specifically tailored for NMP have been proposed [57] with the aim of increasing resiliency to lost and late data packets. Despite these efforts, several of the currently available NMP frameworks opt for the streaming of uncoded audio data (i.e., $\eta = 1$, $D_c = 0$), thus sacrificing the codec bandwidth savings to avoid additional contributions to the delay budget (e.g. [58]–[60]).

Soundcards often include some form of digital filtering (graphic equalizer, reverberation, etc.). Depending on the implementation, this filtering sometimes introduces additional delay, particularly when implemented in the frequency

[3]Note that η could be either constant or vary according to the specific audio content and codec implementation. For the sake of simplification, in the following we will assume constant compression ratios.

domain through Overlap-and-Add processing (Short-Time Fourier Transform). This, of course, needs to be factored in.

B. SOUNDCARD BLOCKING DELAY AND APPLICATION BUFFERING

General purpose computer architectures do not access the soundcard output on sample-by-sample basis but handle audio data in batches of a given number of samples P (the so called block size), which is typically a multiple of 64. Therefore, before retrieving an audio block, computer processors wait for the generation of P samples. In the meanwhile, the available samples are stored in a physical input buffer, as depicted in Figure 6. One block corresponds to a data volume of PL bits and introduces a blocking delay $D_s = P/R$ s. When the processor accesses the physical buffer by means of a callback function, it copies the P block samples in a second buffer (namely the application buffer) where they will wait to be processed. The choice of P implies a trade-off between system stability and latency: handling larger blocks leads to a more stable behavior w.r.t. small blocks (which impose higher interrupt frequencies and operating system overhead), but more time has to elapse for the generation of the whole batch of samples, which increases the end-to-end latency. In addition, in general purpose operating system there may be multiple processes competing for CPU resources, which could introduce delays in the interrupt triggering process. To overcome this issue, [58] and [61] propose dedicated kernel and network driver solutions. Note that, since the above described functional mechanism holds for both the input and output soundcards, the soundcard blocking delay appears twice in the computation of the OOSE delay.

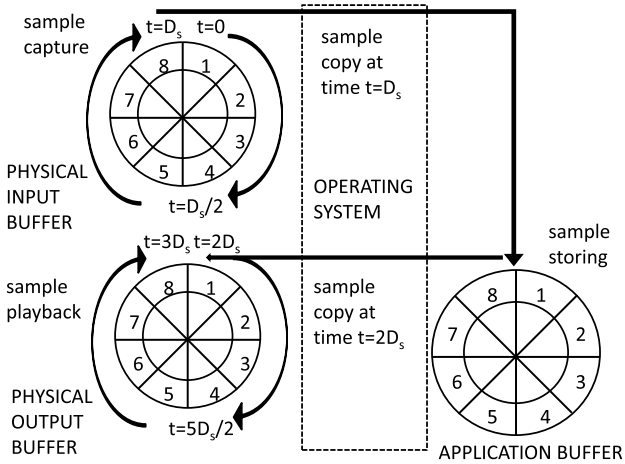

FIGURE 6. Soundcard input/output buffer, assuming that the block size is $P = 8$ samples and that the application buffer introduces no additional queuing time.

Moreover, the application buffer may introduce an additional delay if the application imposes a threshold on the minimum number of queuing blocks before starting the playout. This lag is usually introduced at the receiver side to compensate for the effects of network jitter and clock drift,

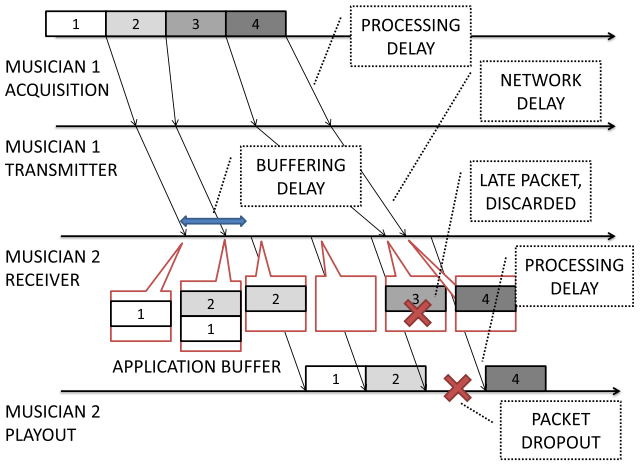

FIGURE 7. Example of packet dropout due to application buffer under-run.

creating enough elasticity so that slight variations of the packet arrival rate are more unlikely to cause buffer under-run conditions (i.e. the application buffer is found empty when the callback function accesses it to copy one block into the soundcard physical output buffer). However, the reverse problem may also occur in case of bursts of packet arrivals or if the clock of the remote side runs faster than the local clock: the receiver buffer may not be able to accommodate all the incoming audio data and some of them have to be discarded (buffer over-run). In turn, buffer under/over-runs lead to artifacts and glitches in the reconstructed audio signal (i.e. micro-silences or significant amplitude discontinuities within very short time intervals due to missing audio samples), as exemplified in Figure 7. A more in-depth discussion on the management of buffer over/under-runs can be found in [62]. Therefore, the application buffer size B needs to be properly sized according to the delay tolerances, possibly with automatic dynamic adjustments to adapt to the varying network conditions (implemented e.g. in [63]). For a comparative evaluation on the latency-audio quality trade-off in different audio streaming engines and NMP frameworks, the reader is referred to [62]. As a rule of thumb, assuming that the application at the receiver side waits until a half of the buffer size is filled with data before starting the playout, the application buffer delay can be estimated as $D_b = \frac{BP}{2R}$.

C. PACKETIZATION DELAY

Once the media content is avaliable at application level at the client side, it can be packetized and transmitted over the telecommunication network. The processing delay taken by the layers of the ISO/OSI stack [64] mainly depends on the machine hardware and is in the order of hundreds of microseconds, thus introducing negligible contributions to the delay budget. During this process, packet headers are added at each layer (from application to physical layer), whose size depends on the specific protocol choices and

implementations and results in an overall overhead of H bits for each audio block.[4] Since H is a constant term and does not depend on the packet size, the smaller the soundcard block size P, the higher will be the number of packets generated in a given time interval and consequently the higher the overhead due to packet header addition (see [65] and [66] for the detailed computations of overhead and overall data rates of ULD-encoded audio data assuming ADSL network access technology). Analogous considerations hold for the reverse de-packetization process at the receiver side, which removes the packet headers layer by layer from the physical to the application level.

D. NETWORK DELAY

The network delay includes three contributions: the transmission delay imposed by the bandwidth C of the network interface card, the propagation time required by the signal to propagate over the physical medium from source to destination, and the processing delay introduced by the network nodes (i.e. switches and routers). The transmission delay depends on the volume of the data stream and can be computed as $D_t = \frac{R\eta ChL + \lceil R\eta ChL/P \rceil H}{C}$.

Conversely, the propagation delay depends exclusively on the choice of the transmission medium: being c the speed of light, the propagation speed c_m is roughly $0.8c$ for copper cables and $0.7c$ for optical fibers (which are the typical medium used in long-haul backbone networks). Therefore, the propagation delay results to be in the order of 5 μs/km and can be calculated as $D_p = \mathcal{L}/c_m$, where $\mathcal{L}$ is the physical distance between source and destination.

Finally, the processing delay D_q introduced by each intermediate network node is a function of multiple factors including the network topology and traffic congestion conditions, the implemented routing algorithms, the policies applied by network operators to ensure quality of service guarantees, the queuing mechanisms adopted by routers and switches [67]. Analytic models taking into account all the above listed features cannot be obtained except for extremely simple network configurations. Therefore, the approach adopted by a substantial body of literature is to model the overall network delay (thus including propagation, transmission and processing delays), as a random variable with a given statistic distribution, whose parameters are adjusted according to the network characteristics (among the numerous studies, see e.g. [68], [69]).

E. ESTIMATION OF OOSE DELAY

Summing all the above discussed contributions, the OOSE delay experienced by the user of a NMP system can be estimated as:

$$D_{tot} = 2(D_a + D_c + D_s) + D_b + D_p + D_t + D_q \quad (3)$$

[4]Note that a single block may be split over multiple packets at lower layers due to the restrictions imposed on the maximum packet size. When computing H, block splitting must therefore be taken into account.

where the term $D_a + D_c + D_s$ is considered twice to account for the soundcard delay contributions at both transmitter and receiver side. Assuming that no information about D_q is available and that the bandwidth B is sufficiently high to make the impact of the transmission delay negligible, D_{tot} can be lower-bounded by the amount:

$$D_{tot} \geq 2\left(\frac{1+F+P}{R} + D_c\right) + \frac{BP}{2R} + \mathcal{L}/c_m \quad (4)$$

Apart from the propagation delay $\mathcal{L}/c_m$, which can be estimated with a rough computation of the geographical distance between the NMP players, and from the coding delay D_c, which can be varied or even eliminated depending on the choices about the audio codec usage, the remaining contributions exhibit an inverse dependency on the audio card sampling rate R and are directly proportional to the audio block size P, as depicted in Figure 8.

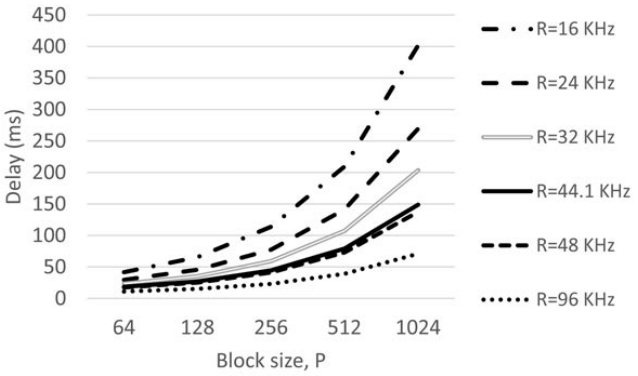

FIGURE 8. Dependency of the delay lower bound on the audio block size and soundcard sampling rate, assuming a soundcard filter length of $F = 100$ samples, an application buffer size of $B = 8$ blocks and a geographical distance of $\mathcal{L} = 1000$ km.

F. PACKET LOSS CONCEALMENT TECHNIQUES

When a packet arrives corrupted at the receiver, or arrives too late to be able to contribute to the audio stream, or simply does not arrive at all, actions must be taken in order to minimize the impact of the missing information. The literature is rich with techniques for containing the damage, involving the receiving end as well as the sender, which in some cases are specifically designed around the coding format [70], and in other cases focus on signal reconstruction. In this Section we offer a very brief summary of such solutions [71]–[73].

The need of successfully repairing packet losses tends to be in contrast with the requirements of real-time low-latency operation, which are typical of NMPs, so in some cases [58] the approach consists in not doing any correction, assuming the network service is so reliable that packet losses occur with negligible probability (e.g. in the order of $10^{-6} - 10^{-8}$ with network jitter below 1 ms, as in academic networks).[5]

Alternatively, the simplest solution on the sender side consists of transmitting duplicate packets in order to reduce

[5]http://www.garr.it/eng

the probability of losing them, but this tends to increase the data rate in the stream and, consequently, worsen the delay. Another sender-based countermeasure against packet loss implies packet interleaving [74], which is done with the purpose of dispersing data vacancies, thus limiting their size to one or two packets at a time. This is an operation that most audio compression schemes can do without additional complexity, but the price to pay is an added delay that depends on the interleaving factor in packets, which is there even if no repair is needed. Again, on the sender side, it is possible to add redundancy data to the stream, which can be used on the receiver side to repair data losses and recover lost packets. Such solutions, are known as Forward Error Correction (FEC) methods [71], can be classified into media-independent or media-dependent. The former do not consider what information is being sent in the packets, while the latter take into account whether it is an audio signal or a video signal. Media independent techniques are suitable for both audio and media content and do not depend on which compression scheme is being considered. Furthermore, their computational cost is limited and are easy to implement. On the other hand, they tend to worsen the delay (repair cannot begin until a sufficient number of packets is collected). Furthermore, they tend to increase the bandwidth requirements at the risk of causing additional packet loss. Media-specific FEC methods, on the other hand, are characterized by low added latency, as they usually only need to wait for a single packet to repair (if dealing with burst-like losses, the latency tends to worsen). Furthermore, they do not significantly increase the transmission rate, in comparison with media-independent FEC solutions. However, computational cost is a factor as well as implementation complexity, which may impact on the final quality.

If we focus on the receiver side, we talk about Packet Loss Concealment (PLC) strategies. In PLC the receiver does the best it can to recover data losses, and it works best for limited packet loss rate ($<$ 15%) and for small packets (4-40 ms). Such solutions generally perform less well than sender-based methods and therefore they are not usually employed. Because of their limited effectiveness, it is advisable to use them in conjunction with sender-based methods.

We can identify three wide categories of PLC: those based on data insertion, those that perform data interpolation, and those that rely on data regeneration [71].

Data insertion simply consists of replacing a lost packet with data filler. The simplest solution of the sort is known as zero insertion (or silence substitution) with obvious interpretation of its meaning. This solution has the advantages of preserving the timing of the audio stream and of being of immediate implementation. Notice that a silence of a few ms will be perceived more of as a ''click'' than as a silence in the usual sense, therefore this method will work acceptably well only for very short packets (typically 4-16 ms), while for packets of more standard size (40ms) it will be ineffective. Furthermore, the quality of PLC will start dropping significantly for packet loss rates that are higher than 2%. An alternate solution to zero insertion is noise substitution, which relies on the fact that a certain amount of repair work can be performed by the listener's brain (phonemic restoration) if the data loss is replaced by something other than silence. Typical choices for data fill-ins are white noise or ''comfort noise''. This solution shares similar advantages to zero replacement (preservation of timing, low complexity), but it requires a careful noise magnitude adjustment. Instead of filling the gap with a random signal, we could opt for patching it with a signal excerpt picked from some other location in the audio stream. One straightforward way to do so consists of repeating the previous frame (a.k.a. ''wavetable'' mode, since a continuously repeated packet will create a tone with a fundamental period equal to the buffer size). Better solutions minimize overlapping artifacts through some form of dynamic realignment based on synchronous OverLap and Add (OLA) or Pitch-Synchronous OLA [75]. Such methods rely on a quasi-periodic behavior of the audio stream or some pitch-related peculiarities of the waveshape. Whatever the choice, this splicing operation needs to be done in such a way to seamlessly blend the patch on either side through either direct or synchronized cross-fading. This class of solutions are quite effective as long as the gaps are quite narrow, and the quality significantly drops when the length of the lost packets extends beyond 20ms and/or the packet loss rate grows above 3%. Other limitations of OLA/PSOLA methods are in that they tend to interfere with delay buffering and do not preserve timing.

Data interpolation methods operate by bridging the gap based on the content on either side of it. The simplest method of the sort is *waveform substitution*, which consists of waveform repetition or mirroring [76] from both sides of gap. This represents an improvement over simple repetition, which uses information from just one side of the gap. PLC methods based on waveform substitution are particularly popular thanks to their implementational simplicity. One such method is also proposed in ITU recommendation G.711 Appendix I [77]. There are also hybrid solutions based on pitch waveform replication, which rely on repetition during the unvoiced portion of the signal (typically speech) and extend the duration of the voiced portion of the signal in a model-based fashion. Such solutions tend to perform slightly better than simple waveform substitution. Model-based PLC methods rely on a specific model for patching signal gaps. A very popular choice is the Auto-Regressive model combined with Least Squares minimization (LSAR) [78]. Such methods, however, are only effective for filling the gap left by very short individual packets. An alternative solution consists of relying on *time scale modification*, which *stretches* the audio signal across the gap. This approach generates a new plausible waveform that smoothly blends across the gap. It is computationally heavier, but tends to perform better than other solutions.

Data Regeneration methods use the knowledge of the adopted audio compression technique to derive codec parameters for repair. There are quite a few solutions of the

TABLE 3. Summary of NMP frameworks.

Authors	Name	Architecture	Network range	Network protocols	Supported data type	Nr. of Audio Channels	Multi-stream synchronization	Codec
Saputra *et al.* [91]	BeatME	Client-Server	LAN, WLAN	UDP or OSC [104]	MIDI	16 (input), 1 (output).	none	uncompressed
Kurtisi, Gu *et al.* [86], [104]	-	Client-Server	LAN	RTP,UDP (stream) TCP (session data)	audio	n.a.	NTP	ADPCM, FLAC (real-time) or MP3, MPEG4 (on-demand)
Renwick *et al.* [92]	Sourcenode	Client-Server	LAN	UDP	MIDI	n.a.	none	uncompressed
Stais *et al.* [98]	-	Client-Server or P2P	WAN	n.a.	audio	2	NTP	uncompressed
Kapur *et al.* [85]	Gigapopr	Client-Server	WAN	UDP	audio, video, MIDI	n.a.	n.a.	uncompressed
Wozniewski *et al.* [97]	Audioscape	Client-Server	WLAN	n.a.	audio	1 (input), 2 (output)	GPS	uncompressed
Sawchuk, Zimmermann, Chew *et al.* [36]–[38], [41], [80]	-	Client-Server	WAN	RTP/RTSP, UDP	audio, video, MIDI	16	GPS, CDMA	MPEG1-4
Akoumianakis *et al.* [82], [105]	Musinet	Client-Server or P2P	WAN	SIP (signaling), RTP (stream), HTTP (text)	audio, video	any	none	OPUS (audio), H.264 (video)
Carot *et al.* [81]	Soundjack	P2P	WAN	UDP	audio and video	8	external master clock	ULD, OPUS (audio), uncompressed or JPEG video
Drioli *et al.* [58], [106]	LOLA	P2P	WAN	TCP (control) UDP (stream)	audio, video	8	n.a.	uncompressed audio and video
Lazzaro *et al.* [93]	-	Client-Server (control) P2P (media)	WAN, WLAN	RTP/RTCP,UDP (stream), SIP (signaling)	MIDI	16	RTP/RTCP synchronization tool	MPEG4
El-Shimy *et al.* [84]	-	P2P	LAN		audio, video	n.a.	n.a.	
Fischer *et al.* [63]	Jamulus	Client-Server	WAN	UDP	audio	2	none	OPUS
Caceres *et al.* [59], [88], [107]	Jacktrip	Client-Server or P2P	WAN	UDP	audio	any	software-based audio resampling	uncompressed
Akoumianakis *et al.* [60], [108], [109]	Diamouses	Client-Server or P2P	WAN	RTP, TCP/UDP	audio, video, MIDI	any	internal metronome stream	uncompressed audio, MJPEG video
Gabrielli *et al.* [89], [110]–[112]	WeMust	P2P	LAN, WLAN	TCP or UDP	audio, MIDI	12	software-based audio resampling	uncompressed or CELT
Meier *et al.* [90]	Jamberry	P2P	WAN	UDP	audio	2	external master clock	OPUS
Chafe *et al.* [87]	StreamBD	P2P	WLAN	UDP, TCP	audio	any	none	uncompressed

sort [71], which rely on the interpolation of transmitted state, and interpret what state the codec is in. Such solutions are quite effective and tend to reduce boundary effects. On the other hand they are quite expensive from the computational standpoint, and the improvement tends to flatten beyond a certain level of complexity.

V. STATE-OF-THE-ART NMP FRAMEWORKS

Several software and hardware solutions have been developed to support NMP in the last two decades. In this Section we provide an overview of the state-of-the-art software frameworks listed in Table 3, comparing technological characteristics as well as the specific framework purpose (e.g. e-learning, etc.). For a thorough historical perspective on the milestones achieved in the field of NMP from 1965 on, the reader is referred to [79].

A. FRAMEWORK PURPOSE

Though all the reviewed frameworks are aimed at supporting real-time musical interactions with at least audio transport, their implementations vary according to the designers' artistic concept or to cope with technology-dependent issues.

Combined audio and video frameworks [41], [58], [80]–[82] aim at providing an NMP environment supporting e-learning (in which the visual component plays a fundamental role for an effective acquisition of technical and expressive skills) and content delivery to a passive audience (e.g. real-time concert streaming with immersive sound systems with multiple audio channels, advanced spatial audio rendering techniques and high definition video). High-quality video transmission software solutions developed for telemedicine and cinematography (e.g. [83]) have also been employed for

NMP applications, but their typical latencies (above 60 ms) usually exceeds the tolerance thresholds for real-time synchronous musical performances. Motion capture techniques have also been integrated to improve the performers' control over the audio mix of instrument sources, e.g. by automatically adjusting stereo panning and volume of remote audio sources perceived by each player according to his/her orientation and spatial coordinates with respect to the virtual locations of the other performers [84]. Video data can also be integrated in NMP frameworks to provide alternative forms of feedback e.g., visuals and text elements which are dynamically-generated according to the received audio content [85].

In [86], in addition to real-time NMP, on-demand rehearsals are also supported: in this scenario, audio data is recorded in advance, stored in a dedicated repository and delivered upon request. Also [58] provides support for real-time rehearsals in which the whole activity is completely performed live.

Some other frameworks do not include video data [59], [63], [87]–[90] or support only the transmission of MIDI data, thus restricting the choice of the instruments to electronic ones and excluding voice [85], [91]–[93]. Quite often, an independent video channel is used to accompany the low-latency audio framework, for example, using commodity video-conference technologies with the audio turned off [94] or software video-only transport [95]. A 2004 example of the former, [96] combined Tandberg 6000 video codecs with jacktrip for several channels of real-time performer-performer and audience-audience interaction.

A framework for large-scale collaborative audio environments is described by [97] in which mobile users share virtual audio-scenes and interact with each other or with locative audio interfaces/installations and overlaid virtual elements. This is done by combining mobile technologies for wireless audio streaming with sensor-based motion and location-tracking techniques.

B. ARCHITECTURE

Both decentralized peer-to-peer (P2P) and client-server have been used as architectures for NMP systems. As depicted in Figure 9(a), P2P solutions (implemented in [58], [59], [81], [82], [84], [85], [87], [89], [90], and [98]) require each participant in the networked performance to send audio/video data to each of the other players. Therefore, in case of N participants, every user sends/receives $N-1$ streams, which could heavily hinder scalability: since typical audio rates are in the order of magnitude of hundreds of kilobits per second and uplink link capacities for retail users subscriptions reach at most a few Mb/s, a trade-off emerges between number of participants in the NMP session and audio quality, which degrades when lowering the soundcard audio resolution and/or increasing the codec compression rate.

Conversely, in client-server architectures as the ones proposed in [41], [80], [82], [86], [88], [91]–[93], [97], and [98] each player transmits his/her data streams to a central server, which is in charge of mixing the contributions from the N sources in a single stream and to send it back to each participant, as depicted in Fig. 9(b). This way, the bandwidth request of a client is limited to one single stream in both uplink and downlink directions, whereas the server receives/transmits N data streams. This solution removes scalability issues at the client side, but may require significant bandwidth availability and computational resources at the server side, which often requires also specific hardware configurations to avoid the introduction of additional delay contributions. The added path to the server results in added delay between clients when compared to P2P. Looking in more detail, the packets of each media stream received by the network interface cards of the server are passed from the kernel to the application layer, which replicates them according to the user's requests (communicated via a separated control data stream) and passes them back to the kernel for transmission (see Fig. 10(a)). This involves context switching between the kernel and the application, as well as data replication which grows with the number of participants. Both data copying and context switching between kernel and application are well-known sources of delay. To reduce server latency, different architectural options are available: the authors of [61] and [99] compare three alternative approaches. The first one, depicted in Fig. 10(b), is an FPGA-based solution [100] in which the application layer processes only control data (which are not time-critical) whereas the routing procedure of media packets (i.e. reception, copy and transmission) is entirely hardware-handled without kernel intervention. To do so, a table indicating how to treat each packet is maintained in the NetFPGA memory and modified based on the signaling packets received by the application. Moreover, the NetFPGA can rewrite packet headers

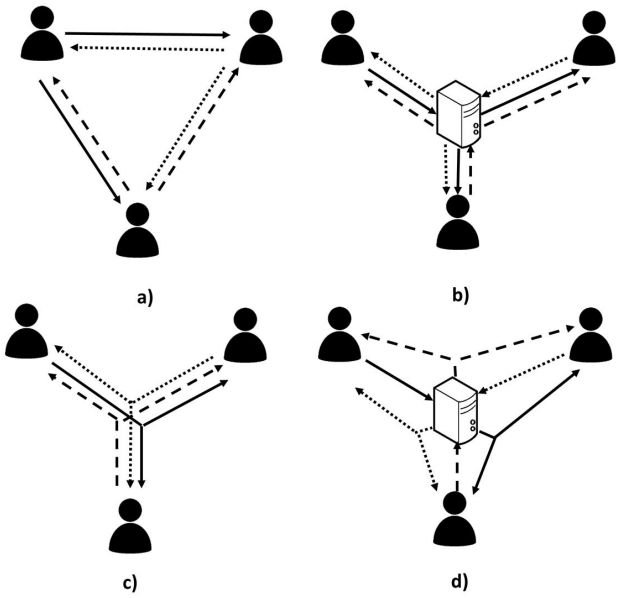

FIGURE 9. Architectures for NMP systems.

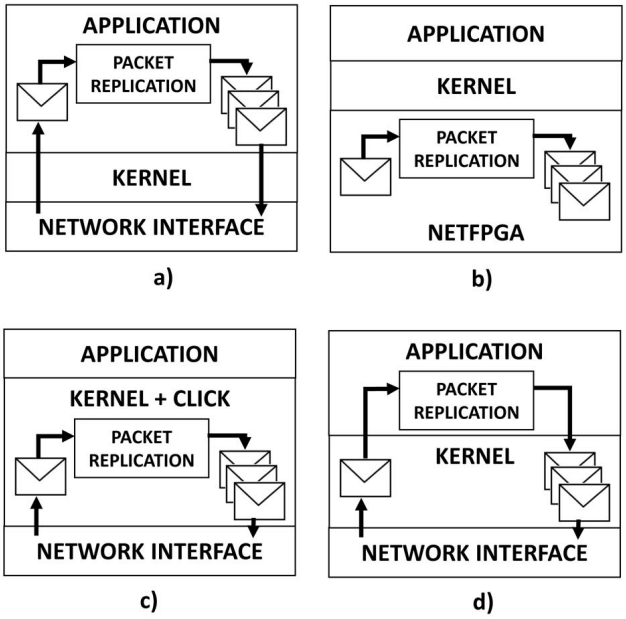

FIGURE 10. NMP server architectural models.

and transmit replicated packets multiple times, thus reducing memory bandwidth requirements. The second solution (see Fig. 10(c)) is a Click [101] modular router which allows for in-kernel execution, thus avoiding most packet copying and context switching. The router can be split in two parts: a control part residing at the application level and a routing part residing at the kernel level. However, the Click router still consumes processor time and cannot perform packet replication without copying. The third solution (see Fig. 10(d)) is a Netmap server framework [102], which enables the application layer to handle the packets directly in the kernel memory without need to copy them in the upper layer, thus avoiding context switching. In this unicast paradigm, each user selectively receives a subset of data streams and can adjust the settings of each of them individually (e.g. volume level, audio/video codec).

Bandwidth consumption issues can be mitigated in case the network natively supports multicast, as in the case of Information Centric Networks [98]: when multicast is enabled, the sender transmits a single copy of the content to be communicated to a pool of receivers, and the network routers duplicate the data when it is needed, according to the network topology and to the addressees' location. Therefore, as depicted in Fig. 9(c, d), in both P2P and client-server infrastructures the uplink bandwidth request in case of multicast is limited to a single data stream (also for the server node [86]). The drawback of the introduction of multicast is that it does not allow for stream customization, since all the players receive the same content.

C. NETWORK SPAN

In principle, a NMP environment should not introduce additional spatial limitations to the geographical displacement of the musicians other than the ones imposed by the signal propagation delays over the physical mediums. As discussed in Section IV, no more than a few thousands of km can be covered to maintain the end-to-end delay experienced by the players below the acceptability thresholds reported in Section III. Most of the frameworks listed in Table 3 are designed to support communications on both Local and Wide Area Networks (WANs). However, the authors of [84], [86], [91], and [92] test their proposed frameworks only on LANs. In other cases (e.g. [58]), the WAN is considered as part of the NMP scenario and a fundamental component; as such, it is supposed to be adequate in terms of bandwidth, latency and error rate to guarantee a sufficient quality of experience. Transmission over Wireless Local Area Networks (WLANs) is supported by [89], [91], and [97], but the latency introduced by wireless communications is typically more pronounced, less controllable and more prone to fluctuations compared to wired mediums. Therefore, in both frameworks the network span is limited to hundreds of meters/few kilometers using Wi-Fi or Bluetooth technologies.

D. NETWORK PROTOCOL STACK

Regarding the protocol stack, all the frameworks listed in Table 3 prefer the usage of UDP [113] as transport layer protocol for audio/video media streaming. UDP introduces less transmission overhead due to the smaller packet-header size compared to TCP and is inherently more suitable for real-time applications due to its lightweight nature, since it does not support any mechanism for packet retransmission, in-order delivery or congestion control. Therefore, packet loss recovery algorithms must be implemented at application level to cope with audio artifacts introduced by missing packets during the media playout. Such algorithms usually combine forward error correction (i.e. the transmission of redundant data in each packet, as in [59]) and error concealment techniques based on data interpolation/substitution [114]. Analogous techniques are applied also to MIDI signals [93]. In one case [58] packet loss recovery techniques are explicitly avoided, to further decrease network latency; in such a case, the network is supposed to be "error free". Some frameworks [82], [86], [93], [104], [108] build data loss management on top of the RTP/RTCP transport protocol [115], taking advantage of the timestamps and sequence numbers included in each RTP packet header. Note that RTP also defines a specific payload format dedicated to MIDI data [116]. UDP with custom sequence numbering schemes have also been used [59]. Frameworks requiring presence discovery of participants, session initialization and the management of textual data or any other type of content in support of the pure media streams rely on SIP and HTTP over TCP, respectively.

E. SUPPORTED DATA TYPES

In NMP, ensuring high audio quality is of great importance in creating acoustical environments providing conditions as close as possible to those of in-presence interactions. Therefore, several frameworks [58], [81], [85], [89], [93],

[97], [98], [108] support the transmission of uncompressed audio data. The authors of [86] opt for the usage of the FLAC lossless codec for real-time performances and of MP3/MPEG4 for on-demand rehearsing: the two latter codifications introduce a startup compression delays of 20 ms or more and are thus considered unsuitable for real-time interactions. Nevertheless, MPEG formats have been used in the framework proposed in [41] and [80]. Alternative solutions rely on low-latency codecs such as the proprietary Fraunhofer ULD [81], CELT [81], [82], [89] and OPUS [81], [90]. The number of supported audio channels varies considerably: most of the frameworks work with mono/stereo configurations, but some of them support channel counts from several to dozens per source [41], [58], [80]–[82], [88], [89], [93].

Conversely, the quality of the video data is less critical for a successful musical interaction and becomes relevant only in case the presence of a passive audience is assumed. In the former case, video codecs such as MJPEG [108], MPEG [41], [80], SVC and H.264 [82] are used, whereas in the latter uncompressed video streaming is supported [58].

F. DATA STREAM SYNCHRONIZATION

When multiple remote locations are involved in a NMP session, the problem of synchronizing audio/video data streams generated by different sources arises, also due to the difference between the nominal and actual value of the audio card hardware clock frequency, which may cause drifts. To this aim, timestamps have to be associated to audio packets and local clocks need a tight synchronization to a common reference to maintain precise timings. The master clock can be transmitted over a side channel or incorporated in the audio streaming data and is reconstructed at the receiver side by means of a Phase-Lock loop. This is accomplished by the RTP/RTCP protocol in [93] or by the Network Time Protocol (NTP) in [86] and [99], with initial synchronization during the NMP session setup phase and periodical refreshments during the session. However, though the accuracy of NTP-based synchronization is about 0.2 ms in LANs, more consistent skews (up to a few ms) may occur in long-distance communications over WANs. Therefore, alternative solutions based on the Global Positioning Satellite (GPS) system and on the time signals broadcasted by the Code Division Multiple Access (CDMA) cell phone transmitters have been investigated in [41], which ensure an accuracy in the order of tens of microseconds. Global time syncronization using affordable, dedicated GPS timeservers [117] have also been considered. A dedicated solution based on the dynamic adjustment of the world reference clock frequency by means of a measurement of the offset between the triggering instants of the remote and the local audio callback functions is discussed in [118] and used in [81] and [90]. The proposed method also implements a jitter compensation mechanism for communications over WANs.

In case of client-server architectures supporting MIDI data [92], synchronization is achieved by means of a MIDI-Clock generated by the server node and transmitted to all the clients, in a master-slave fashion. Clock events are sent at a rate of 24 pulses per quarter note.

All the above mentioned synchronization mechanisms assume that the slave nodes adjust their hardware clock pace according to the clock frequency of the master node. However, software-based solutions relying exclusively on audio resampling have also been investigated: in [112], a digital Infinte Impulse Response (IIR) filter is proposed to estimate the master clock frequency $\hat{R_M}$. Based on such estimation and on the estimated slave local clock frequency $\hat{R_S}$, audio data are resampled at rate R_r defined as:

$$R_r = \frac{\hat{R_M}}{\hat{R_S}} \frac{P_S}{P_M}$$

where P_M (respectively P_S) is the block size at the master (resp. slave) side.

An alternative approach is the propagation of a metronome signal via a dedicated data stream [108]. The signal is generated by the central server (in case of client-server architecture) or by one of the peers (in case of P2P communications).

VI. DISCUSSION

As described in this Overview, networked music performance is an extremely challenging application scenario due to the requirements of this type of interactive communication. What makes it challenging is the fact that musicians are highly sensitive to interaction delays, and in NMP this delay is not just unavoidable but also has a physical lower bound. Many strategies have been adopted for pushing the limits of this type of interactive communication, some aiming at minimizing the time lag in the network, others involving a-posteriori correction strategies. It is only in the past several years, however, that research has begun addressing the problem from a perceptual perspective. We believe this is an extremely promising direction as it is aims at answering open and complex questions that are critical to advancing research in the field.

Research on perceptual aspects of NMP is still in its infancy, and a first significant step ahead could come from assessing under what conditions musicians are able to adapt to NMP limitations. Little is known about adaptability to NMP though a great deal of data is being collected [58] which could help shed light on this aspect. Quite unsurprisingly, direct experience tells us that adaptability is adversely affected by age, but this is not information we can put into productive use unless we sort out the causes. We know that digital ''natives'' (those born just before the turn of the millennium) are often able to immediately adapt to an NMP environment, while older musicians tend to take longer, though it is not clear whether this is to be attributed to a lack familiarity with the devices in use (in-ear headphones, microphones, etc.) or to being less accustomed to interaction delay in mediated communications. A better understanding of these aspects is expected to come from targeted perceptual experiments and data/questionnaire processing.

One aspect that needs to be assessed when discussing perception-aware NMP solutions, is how content influences the quality of the NMP experience. This issue was initially raised in [39], where the tempo slowdowns were found to be dependent on the rhythmic and timbral features of the musical piece that was being played. This aspect indeed deserves further exploration and modeling in order to come to better designed NMP solutions. Content-aware analysis is not easy to achieve because it needs to be approached at various levels of abstraction. The tempo dependency on musical features described in [39], for example, is conducted at a low level of abstraction. Accounting for expressive tempo changes, however, requires content analysis at a higher level of abstraction. Similarly, expressive descriptors are bound to provide important information on the levels of engagement and entrainment of the musicians involved in the NMP, which are likely to play a crucial role in the assessment of the NMP experience.

As mentioned above, understanding and modeling perceptual aspects of NMP requires a large number of perceptual experiments, which are to be conducted in a organized and systematic fashion. One initial outcome of such experiments is a complete revision of all the metrics that are currently adopted for evaluating the quality of NMP experience. The slowing of the tempo, in fact, is one of the most commonly adopted metric for assessing the impact of interactional delay on NMP. In order to reliably assess the quality of the interaction experience, however, we need to account for a much wider range of factors. For example an undesirably limited mutual entrainment between musicians might come into play or, conversely, desirable expressive ritardandi might become part of the performance. Understanding what contributes to the quality of an interactive musical experience, in fact, is still a relatively young and unexplored research problem, which needs to be teased apart from conditions naturally inherent in musical practice. Addressing such issues could greatly help construct models for overcoming, or at least easing, the inherent limitations of NMP.

From the perceptual standpoint, it is also of great importance to explore how different modalities (typically audio and video) jointly contribute to the quality of the NMP experience. For example, there seems to be a clear correlation between video definition/resolution, high frame rate, image size and the overall NMP experience. However, it is still unclear how the NMP experience is affected by such parameters. For example, although we cannot clearly ''see'' the difference between 30 and 90 frames per second in the video refresh rate, the level of comfort is heavily affected by it [119]. Research is only beginning its first step towards understanding the interplay of the various modalities and their parameters towards a better NMP experience.

Beyond perceptual aspects of the NMP experience is the understanding of cognitive mechanisms that musicians develop to cope with NMP limitations, particularly with delays. This understanding would, in fact, lead to novel technological solutions and compensatory measures in NMP. Musicians, over time, learn to adapt to adverse NMP conditions far beyond what we tend to give them credit for. Organ players, for example, may be forced to compensate delays up to over one second between action and perceived sound, when the physical displacement between the instrument keyboard and pipes is significant. Opera singers and choruses are often expected to anticipate their singing of several hundreds of milliseconds with respect to the perceived orchestra sound in order to compensate for the delay caused by their geographic displacement with respect to the orchestra pit. This ability, however, is likely to be associated with a strong musical background and a certain level of proficiency as performer. Is it possible to predict the evolution of the performance ahead enough to compensate for communication delays? Can this prediction ability go as far as anticipating expressive changes in order to preserve the impression of an interactive performance? These questions have recently raised the interest of the scientific community, giving birth to a research field called ''robotic musicianship'' [120].

As we can see, the corpus of knowledge that is yet to be gathered and processed for a better understanding of NMP experiences is, in fact, quite extensive and, as we progress in this research effort, we need to organize it systematically. As done in other fields of research, one effective way to map this knowledge is to develop an ontology that captures the relevant aspects of such knowledge and the relations between them. This requires a careful collection of semantic descriptors and conducting questionnaires for organizing them into semantic spaces equipped with proper metrics. It requires mapping what we know about NMP (and other mediated interactional experiences) into knowledge maps. It also requires using machine intelligence to explicitly map relationships between collected data and semantic qualifiers/descriptors. This is an approach that is commonly adopted in the area of semantic web, or in music information retrieval, but it applies particularly well to all areas of research where knowledge is mapped as a network of relations at various levels of abstraction and strength. We believe that a systematic approach to understanding, modeling and developing solutions for NMP could be extended to other areas of applications where mediated interaction plays a crucial role, particularly gaming and telepresence applications.

Another topic that we believe could be of great interest to musicians and worth considering and exploring, is the possibility to fully exploit the peculiarities of a NMP and turning what are normally considered liabilities of this form of communication into an asset. We know, for example, that NMPs are characterized by

- *freedom from space and ensemble size constraints*: which means that a NMP could involve (in principle) an unlimited number of performers;
- *unbounded augmentation*: the network can transport not just sound but also control signals, therefore a musician could simultaneously play/control a large number of remote musical instruments;

- *virtualization of space*: sometimes in real interactive performances the mutual positioning of musicians is not optimal and not everyone is happy with their mutual spatial location or the environmental conditions; NMP potentially offers the possibility to personalize the positioning and the environment to one's preferences;
- *internet acoustics*: new possibilities open up also on the acoustic forefront, which consist, for example, of sharing a particularly favorable environment with the other musicians and making the whole performance acoustically interact with it; or making several environments become part of a wider shared acoustic space (a global and distributed reverberator), or creating physical model instruments whose delay elements are network delay, or creating a virtual global environment in which all performers interact also from the same acoustical standpoint.

VII. CONCLUSIONS

With this article we offered an aerial view of the current and recent literature on Networked Music Performance research. We discussed how this interaction modality is approached and studied, with special attention to the unavoidable problem of network latency. For this particular aspect we discussed experiments, perceptual aspects and related evaluation metrics, and offered a comparative overview of what constitutes the state of the art. We then discussed hardware/software enabling technologies and experimental frameworks, with a specific telecommunications and networking-oriented focus. Finally we offered a critical perspective on possible future research directions in the field of network-mediated musical interactions.

REFERENCES

[1] A. Barbosa, "Displaced SoundScapes a survey of network systems for music and sonic art creation," *Leonardo Music J.*, vol. 13, pp. 53–59, Dec. 2003. [Online]. Available: http://www.abarbosa.org/docs/barbosa_LMJ13.pdf

[2] T. Blaine and C. Forlines, "JAM-O-WORLD: Evolution of the jam-o-drum multi-player musical controller into the JAM-O-WHIRL gaming interface," in *Proc. Conf. New Int. Musical Express.*, 2002, pp. 1–6. [Online]. Available: http://www.nime.org/proceedings/2002/nime2002_012.pdf

[3] C. Latta, "Notes from the NetJam project," *Leonardo Music J.*, vol. 1, no. 1, pp. 103–105, 1991. [Online]. Available: http://www.jstor.org/stable/1513130

[4] D. Akoumianakis, G. Ktistakis, G. Vlachakis, P. Zervas, and C. Alexandraki, "Collaborative music making as 'remediated' practice," in *Proc. 18th Int. Conf. Digit. Signal Process. (DSP)*, Jul. 2013, pp. 1–8. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6622733

[5] G. Vlachakis, N. Karadimitriou, and D. Akoumianakis, "Using a dedicated toolkit and the cloud to coordinate shared music representations," in *Proc. 5th Int. Conf. Inf., Intell., Syst. Appl. (IISA)*, 2014, pp. 20–26. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6878783

[6] W. Duckworth, "Making music on the Web," *Leonardo Music J.*, vol. 9, pp. 13–17, 1999. [Online]. Available: http://www.jstor.org/stable/pdf/1513470.pdf

[7] P. Rebelo and R. King, "Anticipation in networked musical performance," in *Proc. Int. Conf. Electron. Vis. Arts*. 2010, pp. 31–36. [Online]. Available: http://dl.acm.org/citation.cfm?id=2227180.2227186

[8] C. McKinney and N. Collins, "Yig, the father of serpents: A real-time network music performance environment," in *Proc. 9th Sound Music Comput. Conf.*, 2012, pp. 101–106. [Online]. Available: http://www.chadmckinneyaudio.com/WP-Content/resources/papers/Yigsmc2012.pdf

[9] A. Barbosa, J. Cardoso, and G. Geiger, "Network latency adaptive tempo in the public sound objects system," in *Proc. Conf. New Int. Musical Express.*, 2005, pp. 184–187. [Online]. Available: http://www.nime.org/proceedings/2005/nime2005_184.pdf

[10] W. Woszczyk, J. Cooperstock, J. Roston, and W. Martens, "Shake, rattle, and roll: Getting immersed in multisensory, interactive music via broadband networks," *J. Audio Eng. Soc.*, vol. 53, no. 4, pp. 336–344, 2005. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=13416

[11] C. Alexandraki and I. Kalantzis, "Requirements and application scenarios in the context of network based music collaboration," in *Proc. Conf. I-Maestro Workshop*, 2007, pp. 39–46. [Online]. Available: http://www.academia.edu/5590294/Requirements_and_Application_Scenarios_in_the_Context_of_Network_Based_Music_Collaboration

[12] A. Carôt and C. Werner, "Fundamentals and principles of musical telepresence," *J. Sci. Technol. Arts*, vol. 1, no. 1, pp. 26–37, 2009. [Online]. Available: http://artes.ucp.pt/citarj/article/download/6/5

[13] F. Winckel, *Music, Sound and Sensation: A Modern Exposition*. Chelmsford, MA, USA: Courier Corporation, 2014. [Online]. Available: http://scitation.aip.org/content/aapt/journal/ajp/36/4/10.1119/1.1974536

[14] N. P. Lago and F. Kon, "The quest for low latency," in *Proc. Int. Comput. Music Conf.*, 2004, pp. 33–36. [Online]. Available: http://www.ime.usp.br/~kon/papers/icmc04-latency.pdf

[15] T. Mäki-Patola, "Musical effects of latency," *Suomen Musiikintutkijoiden*, vol. 9, pp. 82–85, Mar. 2005. [Online]. Available: http://www.jaimeoliver.pe/courses/ci/pdf/makipatola-2005.pdf

[16] A. Askenfelt and E. V. Jansson, "From touch to string vibrations. I: Timing in the grand piano action," *J. Acoust. Soc. Amer.*, vol. 88, no. 1, pp. 52–63, 1990. [Online]. Available: http://scitation.aip.org/content/asa/journal/jasa/88/1/10.1121/1.399933

[17] M. Sarkar and B. Vercoe, "Recognition and prediction in a network music performance system for Indian percussion," in *Proc. 7th Int. Conf. New Interfaces Musical Express.*, 2007, pp. 317–320. [Online]. Available: http://doi.acm.org/10.1145/1279740.1279809

[18] C. Alexandrak and R. Bader, "Using computer accompaniment to assist networked music performance," in *Proc. 53rd Int. Conf. Semantic Audio*, 2014, pp. 1–9. [Online]. Available: http://musinet.aueb.gr/papers/AES53_AlexandrakiBader.pdf

[19] B. Vera and E. Chew, "Towards seamless network music performance: Predicting an ensemble's expressive decisions for distributed performance," in *Proc. 15th Int. Soc. Music Inf. Retr. Conf.*, 2014, pp. 489–494. [Online]. Available: http://www.terasoft.com.tw/conf/ismir2014/proceedings/T089_194_Paper.pdf

[20] A. Carôt and G. Schuller, "Towards a telematic visual-conducting system," in *Proc. 44th Int. Conf. Audio Netw.*, 2011, p. 16. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=16137

[21] R. Oda, A. Finkelstein, and R. Fiebrink, "Towards note-level prediction for networked music performance," in *Proc. 13th Int. Conf. New Interface Musical Exp.*, 2013, pp. 1–4. [Online]. Available: http://www.cs.princeton.edu/~roda/pubs/Oda-Finkelstein-Fiebrink_NIME2013.pdf

[22] R. Canning, "Real-time web technologies in the networked performance environment," in *Proc. Int. Comput. Music Conf.*, Sep. 2012, pp. 315–319. [Online]. Available: http://hdl.handle.net/2027/spo.bbp2372.2012.059

[23] M. Ritter, K. Hamel, and B. Pritchard, "Integrated multimodal score-following environment," in *Proc. Int. Comput. Music Conf.*, 2013, pp. 185–192. [Online]. Available: http://hdl.handle.net/2027/spo.bbp2372.2013.023

[24] A. Carôt and C. Werner, "Network music performance-problems, approaches and perspectives," in *Proc. Music Global Village-Conf.*, Budapest, Hungary, 2007. [Online]. Available: http://www.carot.de/Docs/MITGV_AC_CW.pdf

[25] Á. Barbosa and J. Cordeiro, "The influence of perceptual attack times in networked music performance," in *Proc. 44th Int. Conf. Audio Netw.*, 2011, p. 10. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=16133

[26] P. Teehan, M. Greenstreet, and G. Lemieux, "A survey and taxonomy of GALS design styles," *IEEE Des. Test Comput.*, vol. 24, no. 5, pp. 418–428, Sep. 2007. [Online]. Available: http://dx.doi.org/10.1109/MDT.2007.151

[27] M. Gurevich, C. Chafe, G. Leslie, and S. Tyan, "Simulation of networked ensemble performance with varying time delays: Characterization of ensemble accuracy," in *Proc. Int. Comput. Music Conf.*, Miami, FL, USA, 2004, [Online]. Available: https://ccrma.stanford.edu/~cc/pub/pdf/simNetEnsPerf.pdf

[28] C. Chafe and J.-P. Cáceres, and M. Gurevich, "Effect of temporal separation on synchronization in rhythmic performance," *Perception*, vol. 39, no. 7, pp. 982–992, 2010. [Online]. Available: https://blog.zhdk.ch/zmoduletelematic/files/2014/02/temporalSep.pdf

[29] C. Chafe and M. Gurevich, "Network time delay and ensemble accuracy: Effects of latency, asymmetry," in *Proc. Audio Eng. Soc. 117th Conv.*, 2004, pp. 1–7. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=12865

[30] C. Chafe, M. Gurevich, G. Leslie, and S. Tyan, "Effect of time delay on ensemble accuracy," in *Proc. Int. Symp. Musical Acoust.*, vol. 31. 2004, pp. 1–4. [Online]. Available: https://ccrma.stanford.edu/~cc/pub/pdf/ensAcc.pdf

[31] S. Farner, A. Solvang, A. Sæbo, and U. 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. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=15235

[32] P. F. Driessen, T. E. Darcie, and B. Pillay, "The effects of network delay on tempo in musical performance," *Comput. Music J.*, vol. 35, no. 1, pp. 76–89, 2011. [Online]. Available: http://dx.doi.org/10.1162/COMJ_a_00041

[33] C. Bartlette and M. Bocko, "Effect of network latency on interactive musical performance," *Music Percept., Interdiscipl. J.*, vol. 24, no. 1, pp. 49–62, 2006. [Online]. Available: http://www.jstor.org/stable/10.1525/mp.2006.24.1.49

[34] A. Carôt, C. Werner, and T. Fischinger, "Towards a comprehensive cognitive analysis of delay-influenced rhythmical interaction," in *Proc. Int. Comput. Music Conf.*, 2009. [Online]. Available: http://hdl.handle.net/2027/spo.bbp2372.2009.107

[35] A. Olmos *et al.*, "Exploring the role of latency and orchestra placement on the networked performance of a distributed opera," in *Proc. 12th Annu. Int. Workshop Presence*, 2009, p. 9. [Online]. Available: http://astro.temple.edu/~lombard/ISPR/Proceedings/2009/Olmos_et_al.pdf

[36] E. Chew *et al.*, "Musical interaction at a distance: Distributed immersive performance," in *Proc. MusicNetw. 4th Open Workshop Integr. Music Multimedia Appl.*, 2004, pp. 15–16. [Online]. Available: http://www.staff.science.uu.nl/~fleis102/MusicNetwork-IMSC.pdf

[37] E. Chew, A. Sawchuk, C. Tanoue, and R. Zimmermann, "Segmental tempo analysis of performances in user-centered experiments in the distributed immersive performance project," in *Proc. Sound Music Comput. Conf.*, Salerno, Italy, 2005, pp. 1–12. [Online]. Available: http://www.smc-conference.net/smc05/papers/ElanieChew/cstz-smc05_final.pdf

[38] E. Chew *et al.*, "A second report on the user experiments in the distributed immersive performance project," in *Proc. 5th Open Workshop MUSICNETWORK, Integr. Music Multimedia Appl.*, 2005, pp. 1–7. [Online]. Available: http://eiger.ddns.comp.nus.edu.sg/pubs/userexperimentsindip-musicnetwork05.pdf

[39] C. Rottondi, M. Buccoli, M. Zanoni, D. Garao, G. Verticale, and A. Sarti, "Feature-based analysis of the effects of packet delay on networked musical interactions," *J. Audio Eng. Soc.*, vol. 63, no. 11, pp. 864–875, 2015. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=18047

[40] Y. Kobayashi, Y. Nagata, and Y. Miyake, "Analysis of network ensemble with time lag," in *Proc. IEEE Int. Symp. Comput. Intell. Robot. Autom.*, vol. 1. Jul. 2003, pp. 336–341. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1222112

[41] A. A. Sawchuk, E. Chew, R. Zimmermann, C. Papadopoulos, and C. Kyriakakis, "From remote media immersion to distributed immersive performance," in *Proc. ACM SIGMM Workshop Experiential Telepresence (ETP)*, 2003, pp. 110–120. [Online]. Available: http://doi.acm.org/10.1145/982484.982506

[42] S. Hemminger. (2005). *Network Emulation With NetEm*. [Online]. Available: http://developer.osdl.org/shemminger/netem/LCA2005_paper.pdf

[43] B. H. Repp, "Sensorimotor synchronization: A review of the tapping literature," *Psychonomic Bull. Rev.*, vol. 12, no. 6, pp. 969–992, Dec. 2005. [Online]. Available: http://dx.doi.org/10.3758/BF03206433

[44] J. Pressing, "The referential dynamics of cognition and action," *Psychol. Rev.*, vol. 106, no. 4, p. 714, Oct. 1999.

[45] J. P. C. Chomalí, "Synchronization in rhythmic performance with delay [electronic resource]," Ph.D. dissertation, Dept. Music, Stanford Univ., Stanford, CA, USA, 2013.

[46] W. C. Lindsey, F. Ghazvinian, W. C. Hagmann, and K. Dessouky, "Network synchronization," *Proc. IEEE*, vol. 73, no. 10, pp. 1445–1467, Oct. 1985.

[47] H. G. Schuster and P. Wagner, "Mutual entrainment of two limit cycle oscillators with time delayed coupling," *Progr. Theoretical Phys.*, vol. 81, no. 5, pp. 939–945, 1989. [Online]. Available: http://ptp.oxfordjournals.org/content/81/5/939.full.pdf

[48] K. C. Pohlmann, *Principles of Digital Audio*, 5th ed. New York, NY, USA: McGraw-Hill, 2005.

[49] M. Nilsson. (2000). *RFC 3003: The Audio/Mpeg Media Type*. [Online]. Available: https://tools.ietf.org/html/rfc3003

[50] M. Lutzky, G. Schuller, M. Gayer, and U. Krämer, and S. Wabnik, "A guideline to audio codec delay," in *Proc. AES 116th Conv.*, Berlin, Germany, 2004, pp. 8–17. [Online]. Available: http://www.iis.fraunhofer.de/content/dam/iis/de/doc/ame/conference/AES-116-Convention_guideline-to-audio-codec-delay_AES116.pdf

[51] J.-M. Valin, K. Vos, and T. B. Terriberry. (Sep. 2012). *RFC 6716: Definition of the Opus Audio Codec*. [Online]. Available: https://tools.ietf.org/html/rfc6716

[52] J.-M. Valin, G. Maxwell, T. B. Terriberry, and K. Vos, "High-quality, low-delay music coding in the opus codec," in *Proc. Audio Eng. Soc. Conv. 135*, 2013, p. 8942. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=16992

[53] J.-M. Valin, T. B. Terriberry, and G. Maxwell, "A full-bandwidth audio codec with low complexity and very low delay," in *Proc. 17th Eur. Signal Process. Conf.*, Aug. 2009, pp. 1254–1258. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7077384

[54] G. D. T. Schuller, B. Yu, D. Huang, and B. Edler, "Perceptual audio coding using adaptive pre- and post-filters and lossless compression," in *Proc. IEEE Trans. Speech Audio Process.*, vol. 10, no. 6, pp. 379–390, Sep. 2002.

[55] Z. Kurtisi and L. Wolf, "Using wavpack for real-time audio coding in interactive applications," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jun. 2008, pp. 1381–1384. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4607701

[56] A. Pcjić, P. M. Stanić, and S. Pletl, "Analysis of packet loss prediction effects on the objective quality measures of Opus codec," in *Proc. IEEE 12th Int. Symp. Intell. Syst. Inf. (SISY)*, Sep. 2014, pp. 33–37. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6923611

[57] U. Krämer, H. Jens, G. Schuller, S. Wabnik, A. Carôt, and C. Werner, "Network music performance with ultra-low-delay audio coding under unreliable network conditions," in *Proc. Audio Eng. Soc. Conv.*, 2007, p. 7214. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=14272

[58] C. Drioli, C. Allocchio, and N. Buso, "Networked performances and natural interaction via LOLA: Low latency high quality A/V streaming system," in *Proc. 2nd Int. Conf. Inf. Technol. Perform. Arts, Media Access, Entertainment (ECLAP)*, Porto, Portugal, Apr. 2013, pp. 240–250. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-40050-6_21

[59] J.-P. Cáceres and C. Chafe, "JackTrip: Under the hood of an engine for network audio," *J. New Music Res.*, vol. 39, no. 3, pp. 183–187, 2010. [Online]. Available: https://ccrma.stanford.edu/~jcaceres/publications/PDFs/conferences/2009-caceres_chafe-ICMC-jacktrip.pdf

[60] C. Alexandraki and D. Akoumianakis, "Exploring new perspectives in network music performance: The diamouses framework," *Comput. Music J.*, vol. 34, no. 2, pp. 66–83, 2010. [Online]. Available: http://dx.doi.org/10.1162/comj.2010.34.2.66

[61] G. Baltas and G. Xylomenos, "Ultra low delay switching for networked music performance," in *Proc. 5th Int. Conf. Inf., Intell., Syst. Appl., (IISA)*, Jul. 2014, pp. 70–74. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6878798

[62] N. Bouillot and J. R. Cooperstock, "Challenges and performance of high-fidelity audio streaming for interactive performances," in *Proc. 9th Int. Conf. New Inter. Musical Exp. (NIME)*, Pittsburgh, PA, USA, 2009, pp. 135–140. [Online]. Available: http://srl.mcgill.ca/publications/2009-NIME.pdf

[63] accessed on Oct. 1, 2016. [Online]. Available: http://llcon.sourceforge.net/

[64] *Information Technology—Open System Interconnections—Basic Reference Model—The Basic Model*, document ISO/IEC 7498-1, 1996. [Online]. Available: http://www.ecma-international.org/activities/Communications/TG11/s020269e.pdf

[65] A. Carôt and U. Krämer, and G. Schuller, "Network music performance (NMP) in narrow band networks," in *Proc. Audio Eng. Soc. Conv.*, 2006, p. 6724. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=13528

[66] D. van Aken and S. Peckeelbeen, "Encapsulation overhead(s) in ADSL access networks," in *Thomson SpeedTouch*, v1.0 Ed., 2003. [Online]. Available: http://wand.cs.waikato.ac.nz/~513/2006/readings/adsl-2.pdf

[67] L. Angrisani, G. Ventre, L. Peluso, and A. Tedesco, "Measurement of processing and queuing delays introduced by an open-source router in a single-hop network," *IEEE Trans. Instrum. Meas.*, vol. 55, no. 4, pp. 1065–1076, Aug. 2006. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1658355

[68] W. Zhang and J. He, "Statistical modeling and correlation analysis of end-to-end delay in wide area networks," in *Proc. 8th ACIS Int. Conf. Softw. Eng., Artif. Intell., Netw., Parallel/Distrib. Comput., (SNPD)*, vol. 3. Jul. 2007, pp. 968–973. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=4287989

[69] M. T. Lucas, D. E. Wrege, B. J. Dempsey, and A. C. Weaver, "Statistical characterization of wide-area IP traffic," in *Proc. 6th Int. Conf. Comput. Commun. Netw.*, Sep. 1997, pp. 442–447. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=623349

[70] E. Thirunavukkarasu and E. Karthikeyan, "A survey on VoIP packet loss techniques," *Int. J. Commun. Netw. Distrib. Syst.*, vol. 14, no. 1, pp. 106–116, 2015. http://www.inderscienceonline.com/doi/pdf/10.1504/IJCNDS.2015.066029

[71] C. Perkins, O. Hodson, and V. Hardman, "A survey of packet loss recovery techniques for streaming audio," *IEEE Netw.*, vol. 12, no. 5, pp. 40–48, Sep. 1998. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=730750

[72] B. W. Wah, X. Su, and D. Lin, "A survey of error-concealment schemes for real-time audio and video transmissions over the Internet," in *Proc. Intl. Symp. Multimedia Softw. Eng.*, 2000, pp. 17–24.

[73] J.-Y. Lee, H.-G. Kim, and J. Y. Kim, "Packet loss concealment for improving audio streaming service," in *Mobile and Wireless Technology* (Lecture Notes in Electrical Engineering), vol. 310. K. J. Kim and N. Wattanapongsakorn, Eds. Springer, 2015, pp. 123–126. [Online]. Available: http://www.springer.com/us/book/9783662476680

[74] B. W. Wah, X. Su, and D. Lin, "A survey of error-concealment schemes for real-time audio and video transmissions over the Internet," in *Proc. Int. Symp. Multimedia Softw. Eng.*, 2000, pp. 17–24.

[75] J. F. Yeh, P. C. Lin, M. D. Kuo, and Z. H. Hsu, "Bilateral waveform similarity overlap-and-add based packet loss concealment for voice over ip," *J. Appl. Res. Technol.*, vol. 11, no. 4, pp. 559–567, 2013. http://www.sciencedirect.com/science/article/pii/S1665642313715633

[76] K. Maheswari and M. Punithavalli, "Performance evaluation of packet loss replacement using repetititon technique in voip streams," *Int. J. Comput. Inf. Syst. Ind. Manage. Appl. (IJCISIM)*, vol. 2, pp. 289–296, 2010.

[77] *A High Quality Low-Complexity Algorithm for Packet Loss Concealment With G.711*, accessed on Oct. 1, 2016. [Online]. Available: https://www.itu.int/rec/T-REC-G.711-199909-I!AppI/en

[78] J.-H. Chen, "Packet loss concealment based on extrapolation of speech waveform," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Apr. 2009, pp. 4129–4132.

[79] [Online]. Available: http://srl.mcgill.ca/projects/rtnm/history.html

[80] R. Zimmermann, E. Chew, S. A. Ay, and M. Pawar, "Distributed musical performances: Architecture and stream management," *ACM Trans. Multimedia Comput. Commun., Appl. (TOMCCAP)*, vol. 4, no. 2, p. 14, 2008. [Online]. Available: http://doi.acm.org/10.1145/1352012.1352018

[81] A. Carôt and C. Werner, "Distributed network music workshop with soundjack," in *Proc. 25th Tonmeistertagung*, Leipzig, Germany, May 2008, Art. no. 14. [Online]. Available: http://www.carot.de/Docs/TMT08.pdf

[82] D. Akoumianakis *et al.*, "The musinet project: Towards unraveling the full potential of networked music performance systems," in *Proc. 5th Int. Conf. Inf. Intell., Syst. Appl. (IISA)*, Jul. 2014, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6878779&tag=1

[83] P. Holub, J. Matela, M. Pulec, and M. Šrom, "Ultragrid: Low-latency high-quality video transmissions on commodity hardware," in *Proc. 20th ACM Int. Conf. Multimedia*, New York, NY, USA, 2012, pp. 1457–1460. [Online]. Available: http://doi.acm.org/10.1145/2393347.2396519

[84] D. El-Shimy and J. R. Cooperstock, "Reactive environment for network music performance," in *Proc. New Inter. Musical Exp.*, May 2013, pp. 158–163. [Online]. Available: http://nime.org/proceedings/2013/nime2013_66.pdf

[85] A. Kapur, G. Wang, P. Davidson, and P. R. Cook, "Interactive network performance: A dream worth dreaming," *Organised Sound*, vol. 10, no. 3, pp. 209–219, 2005. [Online]. Available: http://dx.doi.org/10.1017/S1355771805000956

[86] X. Gu, M. Dick, Z. Kurtisi, U. Noyer, and L. Wolf, "Network-centric music performance: Practice and experiments," *IEEE Commun. Mag.*, vol. 43, no. 6, pp. 86–93, Jun. 2005. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1452835&tag=1

[87] C. Chafe, S. Wilson, A. Leistikow, D. Chisholm, and G. Scavone, "A simplified approach to high quality music and sound over ip," in *Proc. COST-G6 Conf. Digit. Audio Effects (DAFx)*, Dec. 2000, pp. 159–164. Available: https://ccrma.stanford.edu/randal/publications/2000 DAFx.pdf

[88] J.-P. Cáceres and C. Chafe, "Jacktrip/soundwire meets server farm," *Comput. Music J.*, vol. 34, no. 3, pp. 29–34, 2010. [Online]. Available: http://dx.doi.org/10.1162/COMJ_a_00001

[89] L. Gabrielli and S. Squartini, "Wireless networked music performance," in *Wireless Networked Music Performance*. Springer, 2016, pp. 53–92. Available:. [Online]. Available: http://www.springer.com/it/book/9789811003349

[90] F. Meier, M. Fink, and U. Zólzer, "The JamBerry—Stand-alone device for networked music performance based on the raspberry Pi," in *Proc. Linux Audio Conf.*, Karlsruhe, Germany, 2014, [Online]. Available: http://lac.linuxaudio.org/2014/papers/6.pdf

[91] R. E. Saputra and A. S. Prihatmanto, "Design and implementation of beatme as a networked music performance (NMP) system," in *Proc. Int. Conf. System Eng. Technol.(ICSET)*, Singapore, Sep. 2012, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6339349

[92] R. Renwick, *SOURCENODE: A Network Sourched Approach to Network Music Performacne (NMP)*. Ann Arbor, MI, USA: MPublishing, Univ. Press, 2012. [Online]. Available: http://quod.lib.umich.edu/cgi/p/pod/dod-idx/sourcenode-a-network-sourced-approach-to-network-music.pdfc=icmc;idno=bbp2372.2012.057

[93] J. Lazzaro and J. Wawrzynek, "A case for network musical performance," in *Proc. 11th Int. workshop Netw. Operating Syst. Support Digit. Audio video*, 2001, pp. 157–166. [Online]. Available: http://doi.acm.org/10.1145/378344.378367

[94] S. Guha, N. Daswani, and R. Jain, "An experimental study of the skype peer-to-peer VoIP system," in *Proc. 5th Int. Workshop Peer-to-Peer Syst. (IPTPS)*, Santa Barbara, CA, USA, Feb. 2006, pp. 1–6.

[95] W. Taymans, S. Baker, A. Wingo, R. S. Bultje, and S. Kost, *Gstreamer Application Development Manual (1. 2. 3)*. Publicado en la Web, 2013.

[96] L. Handberg, A. Jonsson, and K. Claus, "Community building through cultural exchange in mediated performance events: A conference 2005," School Commun., Södertörn Univ., Huddinge, Sweden, Tech. Rep., 2005.

[97] M. Wozniewski, N. Bouillot, Z. Settel, and J. R. Cooperstock, "Large-scale mobile audio environments for collaborative musical interaction," in *Proc. 8th Int. Conf. New Inter. Musical Exp. (NIME)*, 2008, p. 13. [Online]. Available: http://sheefa.net/zack/publications/mobileAudioscape_NIME2008.pdf

[98] C. Stais, Y. Thomas, G. Xylomenos, and C. Tsilopoulos, "Networked music performance over information-centric networks," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC)*, Jun. 2013, pp. 647–651. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6649313

[99] G. Xylomenos, C. Tsilopoulos, Y. Thomas, and G. C. Polyzos, "Reduced switching delay for networked music performance," in *Proc. Packet Video Workshop (Poster Session)*, 2013, pp. 1–2. [Online]. Available: http://mm.aueb.gr/publications/2013-MCU-PV.pdf

[100] J. W. Lockwood *et al.*, "Netfpga–an open platform for gigabit-rate network switching and routing," in *IEEE Int. Conf. Microelectron. Syst. Edu. (MSE)*, Jun. 2007, pp. 160–161. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4231497

[101] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The click modular router," *ACM Trans. Comput. Syst. (TOCS)*, vol. 18, no. 3, pp. 263–297, 2000. [Online]. Available: http://doi.acm.org/10.1145/354871.354874

[102] L. Rizzo, "Netmap: A novel framework for fast packet I/O," in *Proc. USENIX Annu. Tech. Conf.*, 2012, pp. 101–112. [Online]. Available: http://dl.acm.org/citation.cfm?id=2342821.2342830

[103] *Open Sound Control, an Enabling Encoding for Media Applications*, Accessed on Oct. 1, 2016. [Online]. Available: http://opensoundcontrol.org/introduction-osc

[104] Z. Kurtisi, X. Gu, and L. Wolf, "Enabling network-centric music performance in wide-area networks," *Commun. ACM*, vol. 49, no. 11, pp. 52–54, 2006. [Online]. Available: http://doi.acm.org/10.1145/1167838.1167862

[105] D. Akoumianakis *et al.*, "The musinet project: Addressing the challenges in networked music performance systems," in *Proc. Int. Conf. Inf. Intell. Syst. Appl.*, Jul. 2015, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7388002

[106] G. Davies, *The Effectiveness of Lola (Low Latency) Audiovisual Streaming Technology for Distributed Music Practice*, accessed on Oct. 1, 2016. [Online]. Available: https://www.academia.edu/28770528/The_effectiveness_of_LOLA_LOw_LAtency_audiovisual_streaming_technology_for_distributed_music_practice

[107] C. Chafe, "Living with net lag," in *Proc. 43rd Int. Conf. Audio Eng. Soc. Conf. Audio Wirelessly Netw. Pers. Devices*, 2011, [Online]. Available: https://ccrma.stanford.edu/~/pub/pdf/netLag.pdf

[108] C. Alexandraki *et al.*, "Towards the implementation of a generic platform for networked music performance: The diamouses approach," in *Proc. ICMC Int. Comput. MUSIC Conf.*, Belfast, U.K. 2008, pp. 251–258. [Online]. Available: http://quod.lib.umich.edu/cgi/p/pod/dod-idx?c=icmc;idno=bbp2372.2008.101

[109] D. Akoumianakis, G. Vellis, I. Milolidakis, D. Kotsalis, and C. Alexandraki, "Distributed collective practices in collaborative music performance," in *Proc. 3rd Int. Conf. Digit. Interact. Media Entertainment Arts (DIMEA)*, New York, NY, USA, 2008, pp. 368–375. [Online]. Available: http://doi.acm.org/10.1145/1413634.1413700

[110] L. Gabrielli, S. Squartini, E. Principi, and F. Piazza, "Networked beagleboards for wireless music applications," in *Proc. 5th Eur. DSP Edu. Res. Conf. (EDERC)*, Sep. 2012, pp. 291–295. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6532274

[111] L. Gabrielli, S. Squartini, and F. Piazza, "Advancements and performance analysis on the wireless music studio (WeMUST) framework," in *Proc. Audio Eng. Soc. Conv.*, 2013, p. 8896. Available:. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=16796

[112] L. Gabrielli, M. aBussolotto, and S. Squartini, "Reducing the latency in live music transmission with the beagleboard xm through resampling," in *Proc. 6th Eur. Embedded Design Edu. Res. Conf. (EDERC)*, Sep. 2014, pp. 302–306. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6924409

[113] J. Postel. *Rfc768: User Datagram Protocol*. (1980). [Online]. Available: https://www.ietf.org/rfc/rfc768.txt

[114] R. Sinha, C. Papadopoulos, and C. Kyriakakis, "Loss concealment for multi-channel streaming audio," in *Proc. 13th Int. Workshop Netw. Operating Syst. Support Digit. Audio Video*, 2003, pp. 100–109. [Online]. Available: http://www.cs.colostate.edu/~christos/papers/nossdav03.pdf

[115] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. (2003). *Rfc3550: Rtp: A Transport Protocol for Real-Time Applications*. [Online]. Available: https://tools.ietf.org/html/rfc3550

[116] J. Lazzaro and J. Wawrzynek. (2011). *Rfc6295: Rtp Payload Format for Midi*. [Online]. Available: https://tools.ietf.org/html/rfc6295

[117] R. Oda and R. Fiebrink, "The global metronome: Absolute tempo sync for networked musical performance," in *Proc. 16th Int. Conf. New Inter. Musical Exp. (NIME)*, Brisbane, Qld, Australia, Jul. 2016, pp. 11–15.

[118] A. Carôt and C. Werner, "External latency-optimized soundcard synchronization for applications in wide-area networks," in *Proc. AES 14th Regional Conv., Tokio, Jpn.*. vol. 7, 2009, p. 10. [Online]. Available: http://www.carot.de/Docs/AES_Tokyo.pdf

[119] A. Wilkins, J. Veitch, and B. Lehman, "Led lighting flicker and potential health concerns: Ieee standard par1789 update," in *Proc. IEEE Energy Convers. Congr. Exposit.*, Sep. 2010, pp. 171–178.

[120] M. Bretan and G. Weinberg, "A survey of robotic musicianship," *Commun. ACM*, vol. 59, no. 5, pp. 100–109, 2016.

CRISTINA ROTTONDI received the master's (*cum laude*) and Ph.D. (*cum laude*) degrees in telecommunications engineering from the Politecnico di Milano in 2010 and 2014, respectively. She is currently a researcher with the Dalle Molle Institute for Artificial Intelligence in Lugano, Switzerland. Her research interests include cryptography, communication security, design and planning of optical networks, and networked music performance.

CHRIS CHAFE is a Composer, Improvisor, and Cellist, developing much of his music alongside computer-based research. He is a Director of the Center for Computer Research in Music and Acoustics (CCRMA), Stanford University. At IRCAM, Paris, and The Banff Centre, Alberta, he pursued methods for digital synthesis, music performance, and real-time internet collaboration. CCRMA's SoundWIRE Project involves live concertizing with musicians the world over.

CLAUDIO ALLOCCHIO was a Researcher with the Astronomic Observatory, Trieste, and the European Organization for Nuclear Research, Geneva. He is currently a Physicist and a Computer Scientist. He contributed to the foundation of the GARR Consortium (the ultrawide-band Italian research and education network), where he is also a Senior Officer. Since 1990, he has been collaborating with the Internet Engineering Task Force, where he is also a Senior Member of the Application Area Directorate. Since 2005, he has been collaborating to the development of low latency audio visual streaming system and organizes the Network Performing Arts Production Workshop every year as a member of the Programme Committee.

AUGUSTO SARTI received the M.S. and the Ph.D. degrees in electronic engineering, from the University of Padua, Italy, in 1988 and 1993, respectively. His graduate studies included a Joint Graduate Program with the University of California at Berkeley, Berkeley. In 1993, he joined the Politecnico di Milano, Milan, Italy, where he is currently an Associate Professor. In 2013, he also joined the University of California at Davis, Davis, as an Adjunct Professor. He has co-authored well over 200 scientific publications on international journals and congresses as well as numerous patents in the multimedia signal processing area. His research interests are in the area of multimedia signal processing, with particular focus on sound analysis, synthesis and processing, space-time audio processing, geometrical acoustics, and music information extraction. He has also involved in problems of multidimensional signal processing, image analysis and 3-D vision. He is an Active Member of the IEEE Technical Committee on Audio and Acoustics Signal Processing, and is in the Editorial Boards of the IEEE SIGNAL PROCESSING LETTERS and of the IEEE TRANSACTIONS on AUDIO, SPEECH and LANGUAGE PROCESSING.