Networked Music Performance over Information-Centric Networks

Charilaos Stais, Yannis Thomas, George Xylomenos and Christos Tsilopoulos
Mobile Multimedia Laboratory, Department of Informatics
Athens University of Economics and Business
Athens 10434, Greece
{stais,thomasi,xgeorge,tsilochr}@aueb.gr

***Abstract*—Information-centric networking (ICN) constitutes an alternative to the conventional, IP-based, internetworking, with information itself being identified rather than the host where it resides. This approach introduces powerful tools and operations for content delivery, such as native support for multicast. Exploiting this native multicast capability is a very promising approach for multimedia applications such as Networked Music Performance (NMP), where a set of musicians located in different places wish to perform together in real time. While conferencing applications traditionally rely on a Multipoint Conferencing Unit (MCU) that receives media streams from each participant and then retransmits a mixed stream to each one, in NMP we would prefer direct communication between the participants, so as to reduce transmission delays and allow each participant to mix the incoming media streams in the desired manner. In this paper we introduce an ICN-based NMP application exploiting native multicast, and compare its performance with both MCU and non-MCU NMP variants, using both unicast and multicast.**

***Index Terms*—Information-centric networks, networked music performance, multipoint conferencing unit, multicast**

I. INTRODUCTION

The proliferation of the Internet and the widespread availability of broadband access to it have paved the way for diverse real-time multimedia applications. *Networked music performance* (NMP) is one of the most demanding applications of this type as, by targeting real-time music delivery to multiple participants, it imposes strict hardware and software requirements in order to provide an acceptable overall *Quality of Experience* (QoE) to participants. The two major QoE concerns that arise in the NMP context are *mouth-to-ear latency*, which is the time interval between the creation of a sound and the remote playback of this sound by a participant, and *transport reliability*, which is the degree of tolerance to lost, corrupted or delayed packets.

Many architectures have been suggested, and even implemented, for NMP, mainly focusing on overcoming the strict limitations imposed by the application's nature, but their applicability to real-world scenarios remains questionable. In this paper, we discuss alternative architectural approaches towards NMP, focusing on the implementation of NMP over *Information Centric Networking* (ICN), which offers some unique advantages due to its support for native multicast. As a result, we can consider NMP designs operating either with or without a *Multipoint Conferencing Unit* (MCU), using either unicast, multicast, or both. These additional options can hopefully make NMP more practical in the wide area.

The remainder of this paper is organized as follows. In Section II we discuss the performance requirements of NMP, focusing on latency and reliability, and present the potential advantages of ICN for NMP. In Section III we review existing NMP architectures and discuss how these solutions can be modified for an ICN environment. In Section IV we present implementations of these solutions over a prototype ICN architecture and evaluate the performance of each option over PlanetLab, so as to evaluate the impact of each design on the feasibility of NMP. We conclude and discuss our plans for future work in Section V.

II. BACKGROUND

A. Delay and reliability requirements

For music performance, the acceptable upper bound of mouth-to-ear latency is considered to be as low as 25 ms [1]. In the NMP context, this latency includes at least i) the delay introduced by sound capture and playback, ii) the transmission delays through the network links, iii) the propagation delays induced by the physical transmission media, and iv) the packet queueing and processing delays in intermediate routers. In practice, in order to avoid excessive bandwidth consumption (e.g. 1.4 Mbps for CD quality stereo), most NMP applications encode and decode the media streams at the sender and receiver, respectively. This reduces the audio bitrate to 128-256 Kbps, at the cost of adding a significant amount of encoding/decoding latency which, even with a very low delay audio codec, cannot be limited to less than 8 ms [2].

Audio transmission exhibits very low error tolerance, a critical issue when best effort networks, such as the Internet, are used. Traffic congestion which leads to packet drops, or bit errors due to physical channel impairments, may cause problems from irritating noise to session interrupts. Due to their real-time requirements, NMP applications typically use the UDP transport protocol, which introduces very low overhead but does not include any error resilience mechanisms. Consequently, to increase robustness to transmission failures, NMP applications have to resort to application layer solutions, such as data redundancy, packet retransmissions or error concealment. Data redundancy is introduced by the transmitter so as to allow error correction to take place at the receiver without feedback to the sender. Data redundancy inflates the required bandwidth and introduces an additional stage of encoding/decoding, thus inducing delays that are unacceptable for NMP. Packet retransmission aims at recovery from lost or corrupted packets based on feedback from the receiver. While the bandwidth consumed is lower than with data redundancy, retransmissions are usually received too late to be effective for NMP. Finally, error concealment is undertaken by the receiver in an attempt to hide the problem rather than fix it [3], which leads to an unavoidable drop in audio quality. Ideally, in order to reduce packet losses an NMP application should either use dedicated links providing a guaranteed *Quality of Service* (QoS) or exploit networks providing differentiated treatment to selected data flows. As discussed below, ICN comes with the advantage of centralized path selection, thus supporting differentiated service treatment in a direct manner.

B. Latency management

While increased bandwidth can simplify loss recovery, it only slightly affects total latency, as it only reduces the transmission delay component. Since latency is the biggest obstacle to NMP, the approaches that have so far been pursued can be classified in three categories. According to [1], the point of distinction is whether the participants act synchronously, pseudo-synchronously or asynchronously. The most challenging category is called *Realistic Jam Approach* (RJA) and refers to real-time multiparty musical interaction, for example, distributed music rehearsal over a network. In RJA all musicians act synchronously and play along, listening to each other's live performance. The second category is called *Latency Accepting Approach* (LAA) and, as its name implies, accepts network delays that would make RJA fail. In LAA musicians play to the music their partners have performed some measures before, thus the performance is considered pseudo-synchronous. Finally, the third category, called *Remote Recording Approach* (RRA), involves music creation without human-to-human interaction and coordination. In RRA the musicians perform in isolation, under the coordination of an atomic director (e.g. a metronome) and send their audio to their partners, which use the timestamps embedded in each packet to place all audio parts in a recorded track. In this paper we discuss the most interesting and provocative category from the network perspective, i.e. RJA.

C. Conferencing vs. NMP

NMP can in principle be achieved with audio conferencing tools since in both cases the users are dispersively located and the information is interpreted by the sense of hearing. However, there are many important differences between conferencing and NMP, as we will explain below, hence there is a need to reconsider the design of NMP applications. Most interactive multiparty network applications utilize a centralized entity to synchronize the peers and manage the service. In a conferencing context, this entity is called an MCU and has the role of relaying media streams between the participants, also implementing media mixing, switching and transcoding [4]. Many NMP applications also adopt this architecture: all participants send their media streams to the server, which may normalize the delay of the individual flows before forwarding them to each participant. In conferencing, the MCU may either mix all received streams to a single one, or select one incoming stream for forwarding, so as to save bandwidth. For example, in a voice conferencing application, a single participant is typically talking at any given time, therefore it makes sense to only send the voice of the current speaker to all listeners. In NMP however, it is desirable to allow each musician to choose an individual channel mix, according to the nature of the instruments involved and the participants' habits. For instance, the drummer of a rock band may want to hear the bass guitar more than the vocals, in order to maintain a tigher integration between these two instruments. As a result, NMP applications should ideally deliver all media streams to each participant, therefore there is no need to use a centralized server for media mixing purposes.

However, the most important difference between conferencing and NMP is in their delay tolerance characteristics. As a rule, during an audio conference only a single participant speaks at any given time, while the rest are passive listeners. Therefore, the interaction between the participants is almost never simultaneous, which allows rather loose synchronization. In RJA-type NMP applications on the other hand, the musicians' operation should be tightly synchronized. This is hard to achieve if a centralized server intervenes between the participants: besides the fact that the server may stretch the network paths between the participants due to its location, packet processing at the server also adds delay to the streams. Therefore, some NMP approaches (such as Soundjack [5]) employ direct communication between the participants. This reduces latency, but is inefficient in terms of network resource consumption in networks where only unicast transmission is supported, like the Internet. In particular, by omitting the central server, each participant would have to transmit its media stream to the $(n-1)$ other participants and receive their $(n-1)$ streams, leading to traffic hotspots and congestion around each participant as the paths of the various media streams converge. In contrast, in a system based on a centralized server which mixes the streams, only a single media stream is required from each participant to the server and back.

D. The PSI architecture

A publish/subscribe architecture consists of three main elements: publishers, subscribers, and an event notification service, otherwise known as a Rendez-Vous network, consisting of *Rendez-Vous Points* (RVPs) [6]. To announce the availability of some content, a publisher that provides that content advertises it to the responsible RVP by issuing a publication message. The subscribers are the content consumers who show their interest in specific content by issuing subscription messages. Some form of identifier indicating the desired content is included in the publication and subscription messages. The RVPs match these messages and arrange for the content to be transferred from publishers to subscribers.

Publish Subscribe Internet (PSI) is an instantiation of such a public/subscribe architecture in a networking context: publishers and subscribers are located at network nodes, exchanging data via publish and subscribe primitives, facilitated by a distributed rendezvous function consisting of many RVPs. Data items are identified by a *Scope Identifier* (SId) and a *Rendezvous Identifier* (RId): the SId identifies a collection of content items and is mapped to the RVP responsible for this particular collection; the RId identifies a specific content item in the collection and is derived from the application issuing the publication message. The scoping mechanism in PSI has been developed with the aim of limiting access to content: a subscriber can only obtain a desired content item if he has access to the scope in which the content has been published. There can be both physical scopes, such as a corporate network, and logical scopes, such as a social network [7]. Scopes can be organized hierarchicaly, with parent-child relationships.

When an RVP receives a subscription message for an (SId, RId) pair for which it has a matching publication message, it communicates with a *Topology Manager* (TM) to inquire about a suitable forwarding path from the publisher to the subscriber. The TM, which is either a service in the same machine or a stand-alone server, holds all the necessary information on the existing topology, i.e. the interconnection between the routers. Therefore, the TM can calculate an appropriate path between the publisher and the subscriber. If multiple subscribers exist for the same publication, then the TM calculates a multicast distribution tree reaching all subscribers. This centralized calculation of paths simplifies the differentiated treatment of applications in a PSI network, something crucial for NMP applications, as we have discussed above.

Each network path generated by the TM is described by a Bloom filter, adopting the approach of LIPSIN [8]. Bloom filters are probabilistic representations of sets where each set element is encoded as a string of zeroes and ones using a number of hash functions. A set is then represented as the logical OR of all its elements. In LIPSIN each network link has a label corresponding to an element in such a set, while an entire path is represented by ORing the labels of all links in the path. Each packet includes in its header the Bloom filter corresponding to its intended path. When a packet arrives at a router, the router looks at the Bloom filter and determines to which of its outgoing links (possibly, more than one) it will have to forward the packet, by performing a logical AND between the label of each link and the included Bloom filter. Note that this technique supports native multicast, since the Bloom filter in the packet header may represent an entire multicast tree, rather than a unicast path. PSI's support for native multicast is perfectly suited to the needs of NMP applications, as we will see below.

III. Implementing NMP over ICN

Implementing an NMP application over an ICN architecture requires an unusual communication model. Instead of establishing connections between endpoints, media streams are published by their producers and subscribed to by their consumers. Specifically, in an NMP system with a centralized server and only unicast capability, each musician publishes a media stream and the server subscribes to all these media

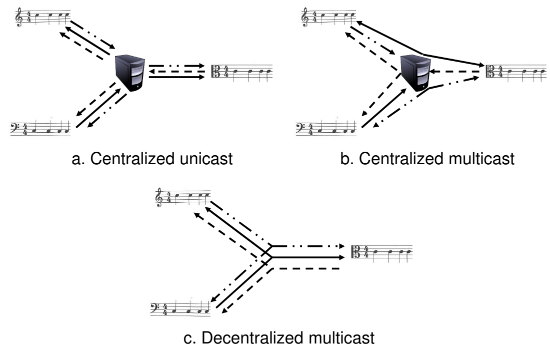

Fig. 1. Different NMP designs over ICN.

streams; in addition, the server publishes a different media stream for each musician, consisting of all media streams except for that musician's stream, and each musician subscribes to that stream. Effectively, the server relays each of the n incoming media streams to the other $n-1$ musicians, as shown in Figure 1.a. If we add multicast capability, then to only change is in the server to musician direction: the server publishes a single media stream, containing all musician's streams, and all musicians subscribe to that stream, as shown in Figure 1.b. Note that in this case musicians also receive their own media streams, hence media packets must include an indication of their origin, so as to allow musicians to drop their own media. Since NMP does not require a centralized server for mixing, when multicast is available we can use instead a decentralized multicast architecture, where each musician publishes a single media stream and subscribes to the streams produced by the other musicians. In this case all streams are distributed via multicast, as shown in Figure 1.c.

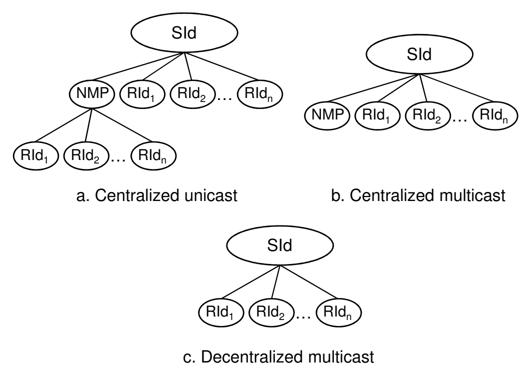

Fig. 2. Information graphs for different NMP designs.

These publish/subscribe requests create an information graph, consisting of an SId and RId hierarchy. In all cases above, we start with a common SId for all participants in the NMP application, as shown in Figure 2, under which each musician publishes its own media stream with a different RId. In the centralized unicast approach (Figure 2.a) the server subscribes to all these RIds and then publishes a customized media stream for each participant using a different RId, below its own NMP scope; each musician subscribes to its own

RId below the NMP scope to receive the streams of other musicians. In the centralized multicast approach the server simply publishes all musicians media streams under a single NMP RId (Figure 2.b), to which all musicians subscribe, thus also receiving their own data. Finally, in the decentralized multicast approach, each musician subscribes directly to the media streams produced by all other musicians (Figure 2.c).

IV. Performance evaluation

A. Experimental setup

In order to compare the different NMP designs presented above, we relied on the Blackadder prototype of the PSI architecture [9], which implements all the functions described in Section II. We extended the functionality of the VoPSI voice conferencing application [10] in order to implement three different NMP designs: one works without a server, exploiting the native multicast capability of the network, while the other two rely on a centralized server receiving all media streams and relaying them to participants via either multicast or unicast. We assumed that we have three musicians located in different places performing for 210 sec (the typical duration of a rock song) and we shipped the data stream across the network in packets with a payload of 512 bytes of data each. As the different NMP variants only differ in their network architecture, in this study we focused exclusively on network delays, hence we simply used uncompressed PCM stereo audio sampled at 8 kHz, producing a bit rate of 128 Kbps, which translates to around 6600 packets per musician. A real NMP application would use higher quality compressed audio, but the overall data rate would be similar.

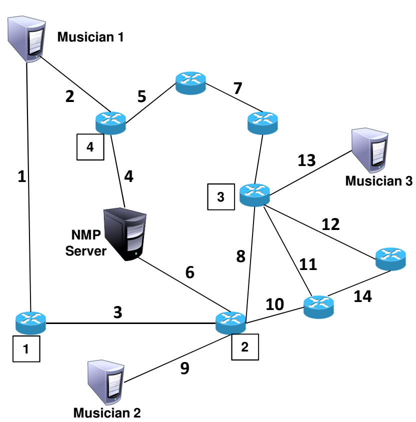

Fig. 3. Topology used for experiments.

We used a PlanetLab slice [11] for the interconnection of the musicians, in order to create a networking scenario with realistic propagation delays and packet delivery latencies, due to the presence of ordinary cross traffic. We deployed the Blackadder prototype on every node of the topology and we placed the domain's TM and RVP alongside the NMP server. The network topology that we created, shown in Figure 3, is similar to an ISP from Rocketfuel [12], Restena [13], the research and education backbone of Luxembourg. The Rocketfuel topology was mapped into the PlanetLab slice

Link	1	2	3	4	6	8	9	13
Delay (ms)	39	19	27	19	28	8	28	28

TABLE I
Link propagation delays.

resulting in a European-wide overlay network where routers 1, 2, 3 and 4 in the figure were located in Finland, France, UK and Italy, respectively, and all musicians were in our lab in Greece. The NMP server was located as close as possible to the center of the topology, in order to avoid unfairness against the centralized approaches. There was no direct connection between the musicians, hence all packets were routed through the PlanetLab overlay. The propagation delays of the network links used to route NMP data are shown in Table I, as measured via the `ping` utility (average of 10 round-trip times, divided by 2); delays are indicative as PlanetLab traffic varies.

The Blackadder prototype transmits raw packets, which are ideal for our experiments, as there are no error resilience mechanisms to burden our measurements with recovery delays. This does not mean that we consider reliability schemes redundant; indeed, reliability enhancement schemes customized for NMP are a promising future research direction. In order to achieve more accurate results, the operating system clocks of the musicians' computers were synchronized during the experiments through the *Network Time Protocol* (NTP) from the same local server. Delays were measured by timestamping packets right before handing them to Blackadder and examining the timestamps right after receiving the packets from Blackadder, hence they only reflect networking induced latencies.

The standard Blackadder TM calculates the shortest path in terms of hops between the publisher and each subscriber; multicast is achieved by merging these shortest paths into a tree. Specifically, in Figure 3, in the decentralized multicast design, data from Musician 1 follow the path $\{1,3\}$ and then split to $\{9\}$ and $\{8,13\}$. In the centralized multicast design, data from Musician 1 firstly travel along $\{2,4\}$ up to the NMP server, and then the NMP server forwards them towards $\{4,2\}$ (back to Musician 1, which discards them) and $\{6\}$, where they split to $\{9\}$ and $\{8,13\}$. In the centralized unicast design, data first get to the NMP server through $\{2,4\}$, and then two separate unicast streams first continue together over $\{6\}$ and then follow different directions ($\{9\}$ and $\{8,13\}$).

B. Experimental results

Our first metric is the average network latency of the packets from all sources to each musician, as measured at the receiver. As shown in Figure 4, the average latency is clearly lower with decentralized multicast, since media do not have to go through the NMP server. On the other hand, the two centralized approaches perform practically the same, given that the delays induced by PlanetLab traffic are expected to lead to such variations between experiments. While the resulting delays are too high for NMP, something inevitable due to the use of the PlanetLab overlay, it is clear that there are delay gains to be made by the decentralized multicast design.

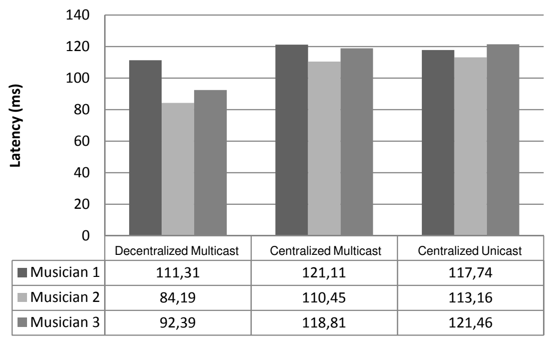

Fig. 4. Average packet latency at each receiver.

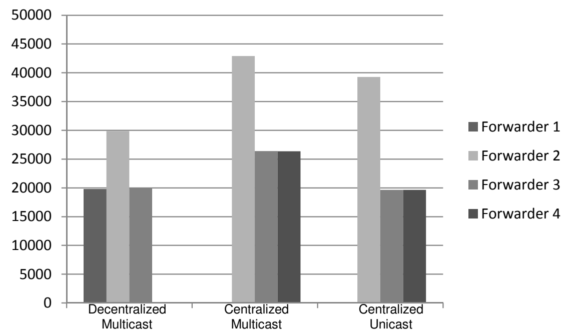

Fig. 5. Number of packets handled by forwarding nodes.

Our second metric is the number of packets handled by the forwarding nodes used in our topology, shown in Figure 5, which reflects the traffic footprint of each experiment; note that each experiment only uses three out of the four forwarders. Rather unexpectedly, the worst performance is exhibited by the centralized multicast design, due to the use of a single multicast stream for all participants: as there are only three participants, one third of the data received by each musician is redundant. Centralized unicast only transmits the right streams to each musician, but the use of unicast and the centralized server may cause the same data to be sent twice over the same links. This is why the traffic footprints of decentralized multicast and centralized unicast are similar except for Forwarder 2, which handles 30% more traffic with centralized unicast. With centralized unicast the same data cross Forwarder 2 twice, for example, the data from Musician 3 to Musician 2 first follow the path $\{13,8,6\}$ to the NMP server and then the path $\{6,9\}$ to the final destination, thus crossing Forwarder 2 twice; the same holds with other paths between participants, due to the need for all media streams to reach the NMP server first.

V. Conclusion and Future Work

In this paper we discussed various designs for NMP and examined how they can be improved by adopting an ICN architecture like PSI. We conducted experiments investigating the effectiveness of the presented approaches, focusing on network latency and traffic load. We found that decentralized multicast is the most promising scheme, with respect to both network latency and resource efficiency. Furthermore, we found that even centralized multicast does not fully exploit the potential of multicast for traffic footprint reduction in an NMP scenario, performing worse than centralized unicast.

This work represents only an initial step within a larger collaborative project on NMP called MUSINET. We are currently working on extending our prototype with ultra low delay audio and video encoding algorithms for high quality media streams, and we will also be experimenting with low latency error recovery techniques in order to improve QoE in lossy environments. In parallel, the project is planning more extensive testing of its prototypes, including measurements of full mouth-to-ear latencies, including capture/coding/decoding/playback delays, as well as experiments with real musicians in order to determine the acceptable limits to latency in NMP.

Acknowledgment

This research has been co-financed by the European Union (European Social Fund - ESF) and Greek national funds through the Operational Program “Education and Lifelong Learning” of the National Strategic Reference Framework (NSRF) - Research Funding Program: THALIS - University of Crete - MUSINET.

References

[1] A. Renaud, A. Carôt, and P. Rebelo, “Networked music performance: State of the art,” in *Proc. of the AES International Conference*, 2007.
[2] J. M. Valin, T. B. Terribery, and G. Maxwell, “A full bandwidth audio codec with low complexity and very low delay,” in *Proc. of EUSIPCO*, 2009.
[3] C. Perkins, O. Hodson, and V. Hardman, “A survey of packet loss recovery techniques for streaming audio,” *IEEE Network*, vol. 12, pp. 40–48, 1998.
[4] M. Willebeek-LeMair, D. Kandlur, and Z.-Y. Shae, “On multipoint control units for videoconferencing,” in *Proc. of LCN*, 1994.
[5] Soundjack. [Online]. Available: http://www.carot.de/soundjack/
[6] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, “The many faces of publish/subscribe,” *ACM Computer Survey*, vol. 35, pp. 114–131, 2003.
[7] G. Xylomenos, X. Vasilakos, C. Tsilopoulos, V. A. Siris, and G. C. Polyzos, “Caching and mobility support in a publish-subscribe Internet architecture,” *IEEE Communications*, vol. 50, no. 7, pp. 52–58, 2012.
[8] P. Jokela, A. Zahemszky, S. Arianfar, P. Nikander, and C. Esteve, “LIPSIN: line speed publish/subscribe internetworking,” in *Proc. of the ACM SIGCOMM*, 2009.
[9] D. Trossen and G. Parisis, “Designing and realizing an information-centric Internet,” *IEEE Communications*, vol. 50, pp. 60–67, 2012.
[10] C. Stais, D. Diamantis, C. Aretha, and G. Xylomenos, “VoPSI: voice over a publish-subscribe internetwork,” in *Proc. of the Future and Mobile Network Summit*, 2011.
[11] Planetlab. [Online]. Available: www.planet-lab.org/
[12] Rocketfuel maps and data. [Online]. Available: http://www.cs.washington.edu/research/networking/rocketfuel/
[13] Restena. [Online]. Available: www.restena.lu