

Exécution répartie de partitions interactives

Jean-Michaël Celerier

Laboratoire Bordelais de Recherche en Informatique, Blue Yeti

Problématique

Étant donnée une partition interactive, on cherche à introduire une sémantique permettant l'exécution de sous-parties de cette partition sur des machines distinctes, de manière à permettre de nouvelles formes d'écriture.

Partitions interactives

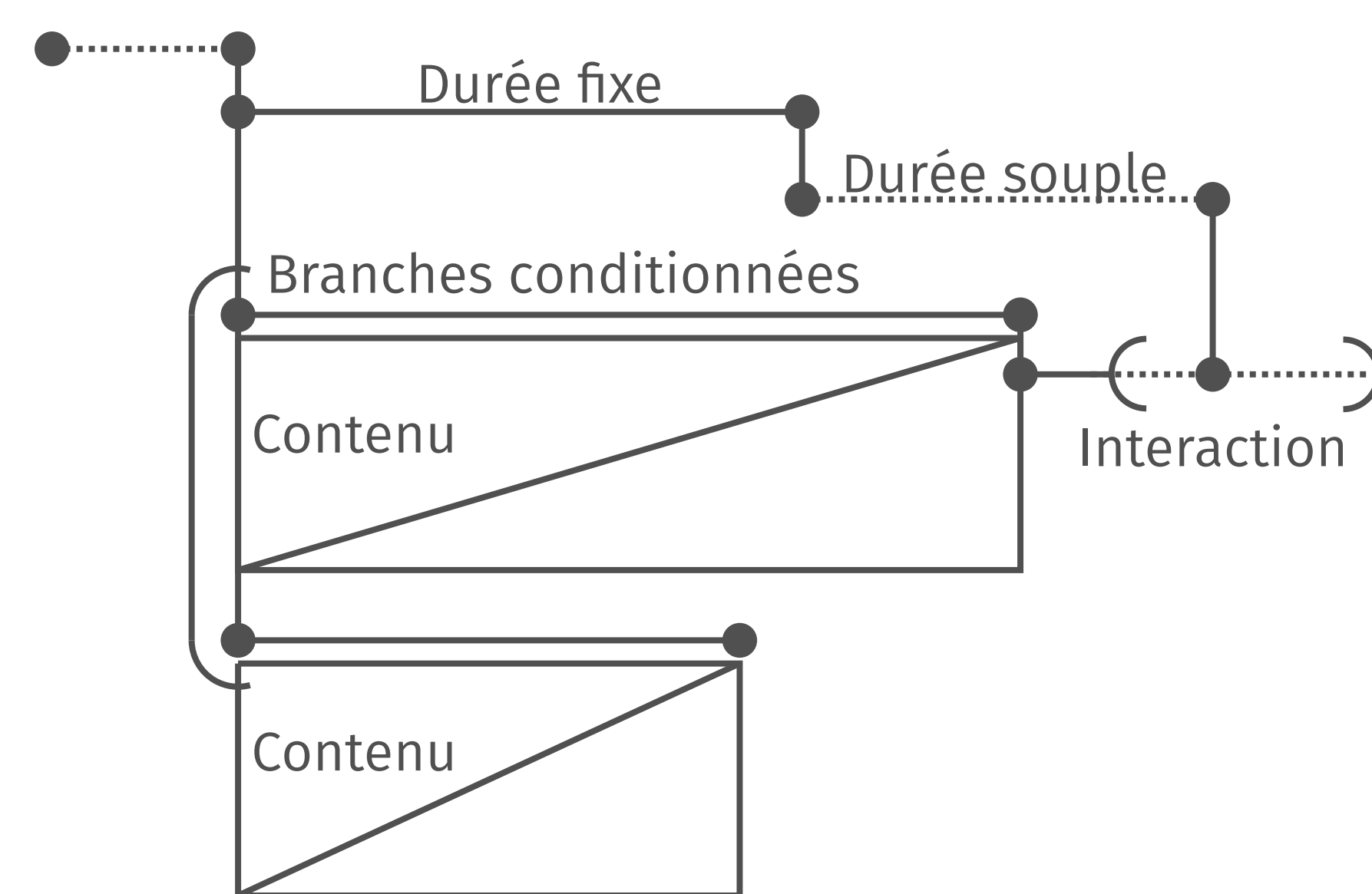


FIG. 1: Syntaxe d'une partition interactive

Un langage de programmation structuré et visuel, axé sur l'organisation temporelle : des primitives permettent l'écoulement du temps, la synchronisation, le bouclage, et la hiérarchisation des contenus. Un moteur Javascript embarqué rend les programmes extensibles.

Il existe des applications pour la musique interactive, la scénographie et le spectacle vivant, ainsi que le contrôle de robots.

Le formalisme propose déjà une répartition bas-niveau. On souhaite rajouter une couche d'écriture permettant de la spécifier explicitement.

Applications visées

- **Murs d'écrans** vidéo.
- Installations artistiques polyphoniques et ouvertes : par exemple permettre d'utiliser la présence de **plusieurs appareils mobiles** lors de l'écriture.
- **Back-up** à chaud des régies de spectacle.
- Réduction de gigue sur périphériques embarqués.

Existant

- **Modélisation algorithmique** : réseaux de Petri pour synchronisation.
- **Horloges** : physiques (NTP[7], PTP[9]), logiques (Lamport[6], vector, matrix) et hybrides[5]. Approches par intervalles plutôt que par dates : Google TrueTime. Horloges adaptées à la gestion du temps musical : Ableton Link, Global Metronome[8].
- **Applications musicales réparties** : Ohm Studio, Splice, Supercollider[3].

Méthode

Abstraction des hôtes réseau sous forme de clients dans des groupes que manipulent les compositeurs. On introduit des annotations de répartition pour les éléments de synchronisation ainsi que pour les structures temporelles.

Répartition des macro-structures

Annotations de groupes sur les éléments de la partition interactive :

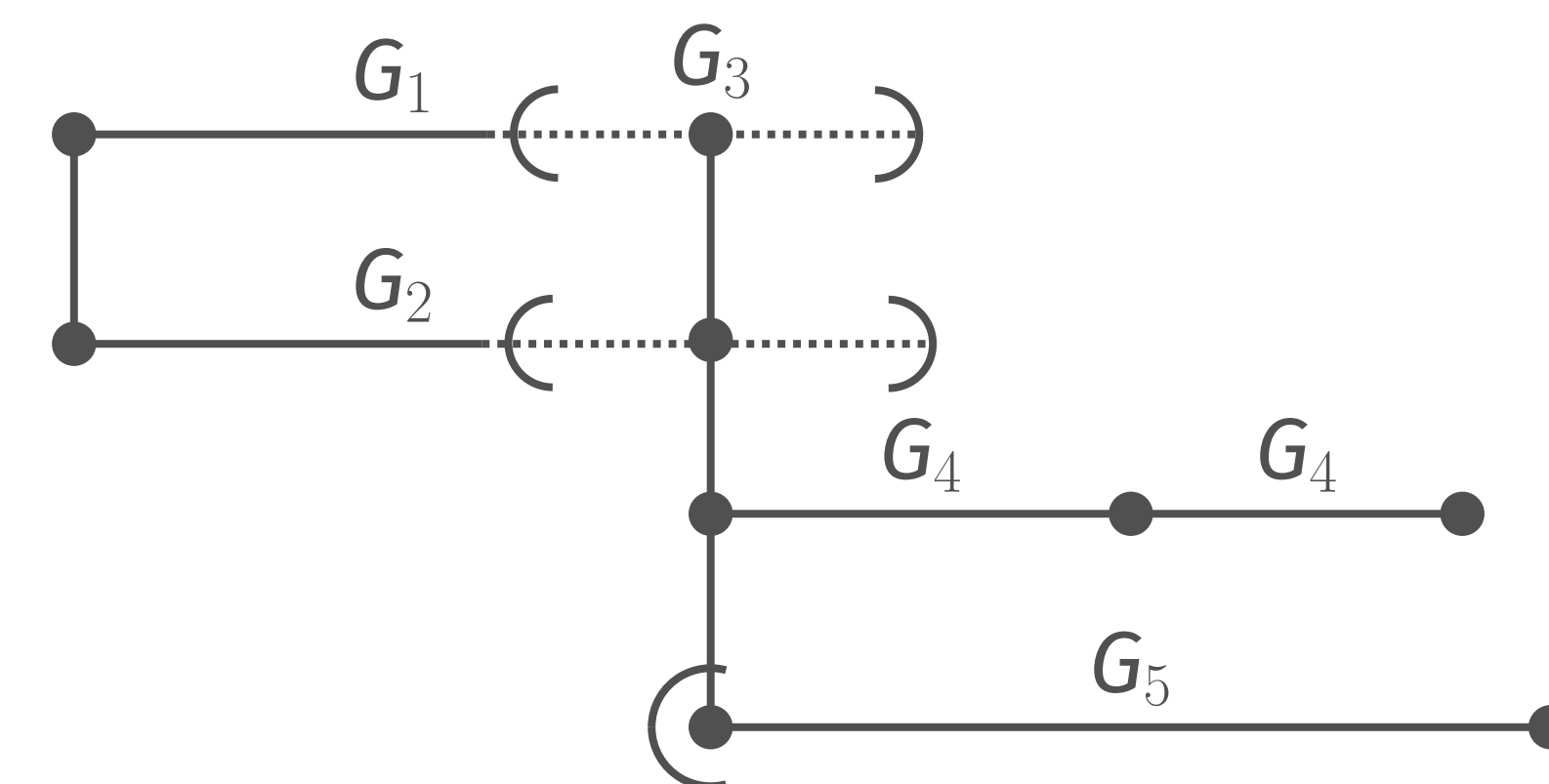


FIG. 2 : Les annotations $G_{1..5}$ s'appliquent récursivement sur les contenus

Trois modes d'exécution :

- **Exécution indépendante** : chaque client du groupe exécute le scénario à son rythme ; les clients des autres groupes ne l'exécutent pas.
- **Partage complet** : toutes les machines partagent la même ligne temporelle récursivement ; chacune exécute les contenus propres à son groupe (fig. 3).
- **Partage mixte** : certaines branches peuvent être exécutées par certains clients et pas par d'autres ; des resynchronisations peuvent avoir lieu en des points donnés par le compositeur.

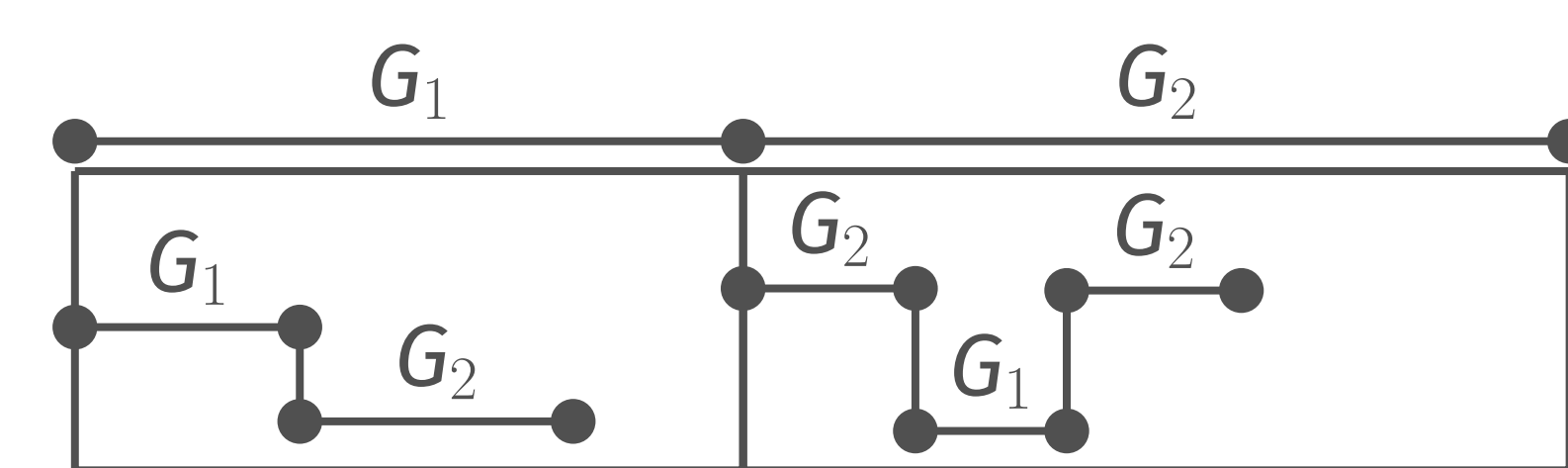


FIG. 3 : Une exécution répartie hiérarchique n'est possible qu'en cas de partage complet : sinon impossible d'assurer la cohérence entre machines.



FIG. 4 : Quarrè (Pierre Cochard, SCRIME, 2017) : une installation en son spatialisé avec plusieurs téléphones.

Sont notamment définis des modes synchrones et asynchrones que peut choisir le compositeur de manière à privilégier la latence, ou bien la synchronisation entre hôtes, en fonction des contraintes de sa pièce.

Synchronisation des micro-structures

Dans le cas d'une exécution partagée d'un scénario, il convient de définir la sémantique d'exécution des points de synchronisation, des conditions, et des vitesses de contraintes, en mode réparti.

C'est fait via deux caractéristiques :

La **synchronisation** :

- **Mode synchrone** : on respecte la sémantique d'exécution des partitions interactives au prix d'une latence de synchronisation.
- **Mode asynchrone** : on ne respecte pas la sémantique avant-après. La fin de l'exécution d'un objet peut survenir quelques millisecondes après le début de l'exécution de l'objet suivant.

Le **délai de propagation** :

- **Propagation instantanée** : dès qu'une information est disponible dans le système, les clients qui en dépendent en sont informés et peuvent y réagir immédiatement.
- **Propagation compensée** : quand une information est disponible dans le système, on estime la date minimale à partir de laquelle tous les clients peuvent être informés en fonction de la latence entre machines. Elles reçoivent toutes un ordre qu'elles sont sensées exécuter à cette date. C'est utile si on désire par exemple une bonne synchronisation entre vidéos.

Via des annotations, le compositeur peut définir le type de répartition qu'il souhaite sur chaque objet.

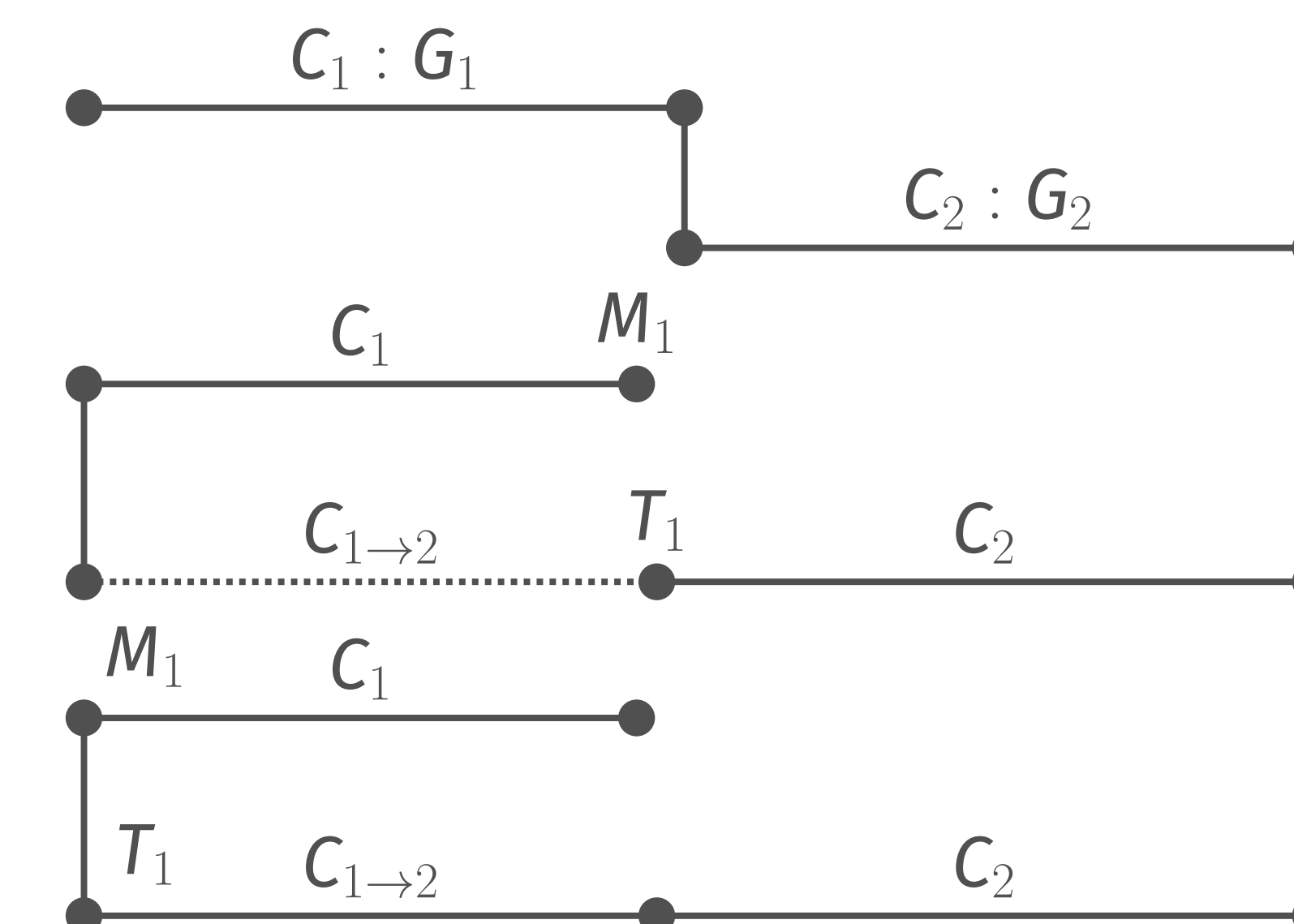


FIG. 5 : Répartition asynchrone précalculée et synchrone compensée d'un scénario avec durées fixes

Résultats

La modélisation de la méthode de répartition se fait elle-même dans le paradigme des partitions interactives. Une implémentation préliminaire, en architecture client-serveur, est offerte en tant qu'extension au logiciel.

Prochains objectifs

- Évolution vers l'édition répartie de partitions interactives, pendant leur exécution. Objectif : permettre à plusieurs régisseurs (son, lumière) de garder la main sur une même exécution.
- Comportements de groupe : utiliser les informations provenant de chaque client, pour générer de nouveaux contenus. Par exemple, étant donné un groupe de plusieurs téléphones portables, pouvoir utiliser les capteurs intégrés et travailler avec le comportement moyen sur ce groupe.

Informations complémentaires

Ce travail a été proposé pour publication aux JIM 2017. Articles sur ce sujet :

- Modèles formels sur lesquels se base le logiciel i-score : [1, 2].
- Paradigme graphique OSSIA : [4].

i-score peut être téléchargé librement sur

- www.i-score.org

Références

- [1] Antoine ALLOMBERT, Gérard ASSAYAG et Myriam DESAINTE-CATHERINE. "A system of interactive scores based on Petri nets". In : *4th Sound and Music Computing Conference (SMC07)*. 2007, p. 158–165. (Visité le 02/11/2015).
- [2] Jaime ARIAS, Jean-Michaël CELERIER et Myriam DESAINTE-CATHERINE. "Authoring and automatic verification of interactive multimedia scores". In : *Journal of New Music Research* (2016), p. 1–19.
- [3] Alexander CARÔT, Pedro REBELO et Alain RENAUD. "Networked music performance : State of the art". In : *Audio engineering society conference : 30th international conference : intelligent audio environments*. Audio Engineering Society. 2007.
- [4] Jean-Michaël CELERIER et al. "OSSIA : Towards a unified interface for scoring time and interaction". In : *TENOR2015*. 2015.
- [5] Sandeep S KULKARNI et al. "Logical physical clocks". In : *International Conference on Principles of Distributed Systems*. Springer. 2014, p. 17–32.
- [6] Leslie LAMPORT. "Time, clocks, and the ordering of events in a distributed system". In : *Communications of the ACM* 21.7 (1978), p. 558–565.
- [7] David L MILLS. "Internet time synchronization : the network time protocol". In : *Communications, IEEE Transactions on* 39.10 (1991), p. 1482–1493.
- [8] Reid ODA, Rebecca FIEBRINK et al. "The Global Metronome : Absolute tempo sync for networked musical performance". In : *New Interfaces for Musical Expression*. Goldsmiths University of London, 2016.
- [9] Yu PENG-FEI et al. "The research of precision time protocol IEEE1588". In : *International Conference on Electrical Engineering*. 2009.