Preliminaries to a network that facilitates phase-based coupling of repetitive human activities

Fred Cummins
Jens Edlund

October 19, 2022

Abstract

We address the problem of how to sustain synchronized repetitive behaviours, such as chanting, over widely distributed networks with very many nodes. Our approach does not try to reduce latency between nodes. Instead, we make use of the repetitive cyclic structure of the behaviour to arrange for synchronization between nodes to be re-cast as phase synchronization with lags between nodes which are integer multiples of the underlying cycle of production. This suggests a novel network topology which we describe formally. The proposed network structure is relatively parsimonious in the number of connections per node, while still ensuring that every node receives productions from every other node, without redundancy. The proposed network topology opens a door to the empirical study of the propagation of (non-temporal) variation through a large network.

1 Introduction

We begin by motivating a novel approach to thinking about how to coordinate widely distributed human behaviours over networks, with a concern for repetitive behaviours. Our ideas are motivated by consideration of how one might organize widespread distributed chanting over networks—a remote goal that eludes current systems (Section 2). We suggest a rethinking of the way time is treated in this problem space, and we adopt a switch from linear time to cyclic time in which phase, rather than seconds, provides a metric (Section 3). In Section 4, then we introduce a novel network topology that implements our proposal for distributed coordination in cyclic time. We restrict ourselves to specifying the topology of the network, which is defined recursively. In the specified network, each node receives transmissions from every other node, with variable time delays. The overall network topology is capable of extension to very many nodes while maintaining parsimony in the number of connections per node. Finally, in Section 5, we return to the motivating question of how to faciliate chant over networks and suggest how a full implementation of the proposed topology might function, and what properties remain to be explored. We wish to suggest that the proposed shift from linear time to cyclic time harbours promise for construction of a novel kind of distributed network suited to supporting repetitive events, such as chanting, but we leave the topic of other possible applications open.

2 A vexing problem: chanting over networks

With the pandemic onset in 2020, the sudden global dependence on Zoom and similar platforms for a vast array of human activities brought some long slumbering issues to the fore. Ingenuity and the force of circumstance allowed novel distributed practices in labour, education and recreation to emerge in a bewildering variety of forms, some highly successful. The resulting and ongoing perturbation to human behaviour has necessarily altered the landscape of research challenges concerned with supporting coordinated activities of all kinds through the mediation of digital networks.

The non-negotiable constraining influence of an airborne pathogen affected different activities in different ways. Some of the obvious and immediate spheres that were hardest hit pertained to those activities in which crowds gather and vocalize together. Sports stadia were emptied, and churches and other sites of religious observance were particularly badly hit. Choirs could no longer function. Musicians became detached from one another. At times, it almost appeared as if a malevolent force were specifically targeting those activities that bring people together to enact their foundational identities. Numerous ingenious and partly successful responses followed, but the unprecedented situation made manifest some long-standing challenges in the development of coordinating technologies that remain unresolved, but that appear now with increased urgency.

We consider here a novel approach to network-mediated support for behaviours that, when done in physical co-presence, involve repetitive and synchronized activity. A key example is provided by joint speech, or chanting, which is a reliable and central feature of practices of worship, sports participation, protest and other foundational human activities. An overview of many of the key features of joint speech are enumerated in Cummins (2018), including the key feature of repetition which recurs in all domains. Because such activities come in a spectacular variety of context-specific forms, we choose to explore a single key challenge that is common to many of them. Our focus is minimal and formal, and its elaboration to support specific real-world elaborations must come later.

The challenge is this. When two or more sound producing nodes attempt to synchronize their productions over networks, even small latencies become immediately destructive. This came to public awareness as people tried, and failed, to chant, sing, make music, or even join in a round of singing “Happy Birthday” over Zoom. The source of the problem is the reciprocity of the joint activity, which decouples the temporality of each producer from that of every other participant. We can describe the “round robin” effect for the simplest case of dyadic production thus:

If A and B attempt to produce an isochronous sequence of vocal events, B’s productions will arrive at A with some latency. The latency may be small (and technological development will continue to shrink such latencies), but they will always, necessarily, be positive. Any variation in B’s timing to which A responds arrives at B with a further latency. A positive feedback loop is thus created ensuring cumulative disruption to the intended homogenous production. The reader is invited to try to sing “Happy Birthday” with a friend over Zoom to illustrate the problem more effectively than this text can.

Previous attempts to improve this situation have focussed on reducing latencies, and much can be done there, but the logic of the decoupling of participants in time is unaffected by such efforts. We choose to thoughtfully consider the problem in a fundamentally different manner. Rather than trying to force together the two sundered temporalities by reducing latency, we begin with the recognition that a fundamental break has occurred. Because of this, we consider only strictly repetitive behaviours in what follows, and we exploit the repetitive nature of the behaviour to recast the goal as alignment, not in linear time, but in phase defined by the cycle of production.

In so doing, we are abstracting away from our original concrete goal of supporting synchronized chanting, and, using an ideal of strictly repetitive behaviour as our guiding image, we set out to explore a network topology that arises in developing that image.

3 A minimal starting point: from linear time to cyclic time

We can illustrate the shift from synchronization based on linear time to synchronization based on cyclic phase with a minimal dyadic example. Two producers, A and B, each produce output such as a short chanted syllable, repetitively, and in isochronous sequence. They are connected by a network that necessarily produces some delay in transmission between producers. Two possible strategies for overcoming lag are illustrated in Fig. 1. Conventional approaches have focussed on reducing lag, and this approach has merit. However we here explore a different strategy that takes advantage of the repetitive isochronous nature of the produced tokens, by witholding reception by B of A's token until one cycle has completed, making it available and perceptible to B at just that point at which the cycle (shared by A and B) repeats.

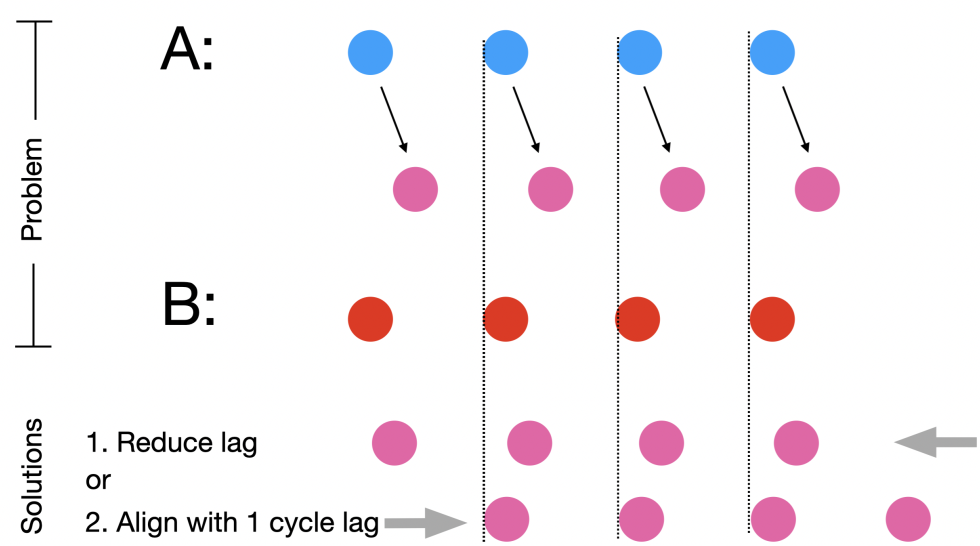

Figure 1: A's productions (blue) arrive at B with some lag (pink). Two strategies can be pursued for overcoming network lag. 1: reduce latency, 2: delay transmission until productions are aligned cyclicly.

With this conceptual shift in place, we now turn to consideration of a network topology to support cyclic coordination among very many producers. In developing the network topology we arrange that transmission from one node to a directly connected node introduces a lag of exactly one cycle. We ignore many practical issues that would arise in implementing a functioning exemplar to focus on the development of a network topology that has the following properties:

- Scalability. The network should allow incremental expansion to very many nodes.
- Connectivity: Each node should receive tokens from every other node at every time step.

- Singularity: The distribution of lags within the array of tokens received by each node is unique to that node, thus ensuring that each node has a distinguished position within the collective.
- Homogeneity: No node has a privileged position within the network when compared with any other node. That is, there are no hierarchies or centers.

We will return to the manner in which a distributed periodicity might be established in implementation in the final section.

4 A novel network topology

We now describe the recursive construction of a network of very many nodes which broadcast and receive tokens in time, that satisfies these conditions.

Each node in the network produces a series of tokens in a homogeneous series, one token per cycle. In order to develop the topology, all nodes are subsumed under the same clock, so that the network as a whole has a fundamental cyclic period, and tokens are synchronized with respect to this cycle. The use of a shared clock driving updates is a simplification for present purposes, in which we ignore imprecision in production and drift in mutual alignment. We return to this issue in the final section. Each node both broadcasts and receives tokens. Passage of tokens across links in the network take one cycle, so that events from one node are received through direct connections with a lag of one cycle.

Our goal is to construct very large networks with manageable numbers of connections per node, yet to ensure that each node "hears" each other node through a unique path, and in a manner unique to its position within the network. We begin with a small set of fully-interconnected nodes, and call this set a cluster. As we then grow the network, additional clusters will be created recursively, and sparse links between clusters will be built, ensuring scalability.

We begin by describing a first order cluster containing three nodes, as in Fig. 2. For expository purposes, we use a constant cluster size of $n = 3$. Any value of n may be used, but for simplicity we assume that n is constant across clusters.

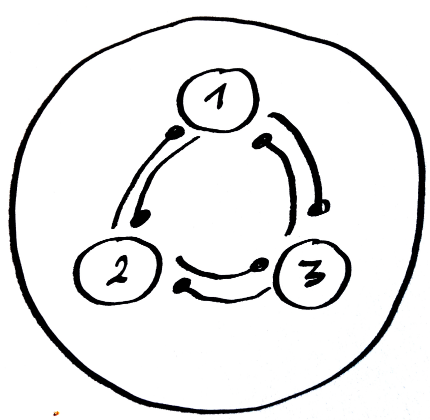

Figure 2: First order network of three fully interconnected nodes

Within a first order cluster, nodes are fully and reciprocally connected by unidirectional links. At each cycle, each node produces one token and broadcasts it to each other node, where it is received one cycle later.

As we construct ever larger networks, we adopt a mode of construction that ensures that the number of per-node connections grows very slowly (linearly), while the number of nodes increases exponentially, all the while ensuring that each node receives tokens from every other node in the network without redundancy or cycles.

We now grow the network by one order. We construct $n - 1$ additional clusters similar to the first, so that the whole network is now a single second order cluster of 3 first order clusters. Indexing of nodes changes as order grows, so that what was Node 1 is now indexed as Node 1.1. Each added cluster (here, 2 clusters have been added) has internal connectivity isomorphic to that within the existing cluster. For connections between clusters, we adopt a sparse approach, such that new connections between the clusters are made between structurally isomorphic positions only. For example, Fig. 3 shows the inter-cluster connections that link to Node 1.1 only. Node 1.1 gains additional connections to Nodes 2.1 and 3.1, but no more. Similar connections link 1.2 to 2.2 and 3.2, and 1.3 to 2.3 and 3.3, and structurally analagous links are created between clusters 2 and 3. Thus when the order of the network as a whole is increased, its number of nodes is multiplied by the cluster size n (here, 3), while the number of additional links generated *per node* increases only additively by n - 1.

For each node, we need to consider both what it receives, and what it broadcasts. In the first order network of Fig. 2, bare tokens are passed between nodes, with lag 1. In the second order network of Fig. 3, broadcasts that span cluster boundaries are composites. (In implementation, these will be additively merged audio streams.) Node 1.1 sends a composite of the productions of 1.1, 1.2 and 1.3 to cluster-external nodes 2.1 and 3.1. Likewise, from 2.1, it receives a composite of 2.1, 2.2 and 2.3. Thus Node 2.1 will recieve all tokens produced within Cluster 1, but in the composite token it receives from 1.1, the production of Node 1.1 will have a lag of 1, while those of Nodes 1.2 and 1.3 will have a lag of 2 (as node 1.1 received these tokens with a lag of 1).

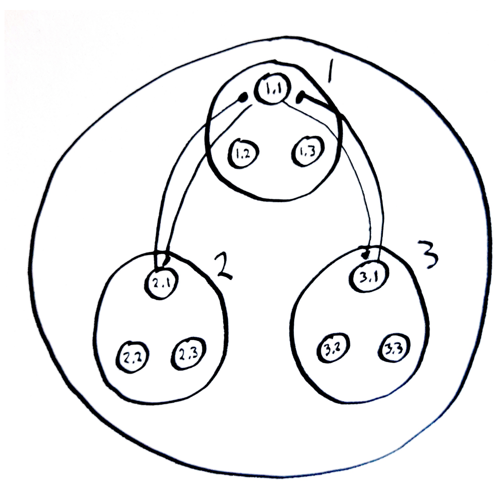

Figure 3: Second order network of three first order clusters. Connections between clusters are made for structurally isopmorphic nodes only. Added connections to Node 1.1 only are shown.

All token passing is from node to node, but connections that span clusters pass along composite tokens that aggregate the productions within the sending cluster. The architecture thus specified implements two important constraints on the flow of tokens within the network.

1. Constraint 1: Within a cluster, all tokens passed are produced within that cluster.
2. Constraint 2: Links from one cluster to another transmit only tokens generated outside the receiving cluster.

Full reciprocal interconnection within first order clusters requires $n-1$ connections per node. By contrast, as higher order clusters are constructed, growth in connections per-node grow much more slowly. For a node within a level 1 cluster, there are $2*(n-1)$ connections to other nodes within the cluster. Adding a single node to the cluster will generate 2 additional connections *per node*, so that each new node comes at a cost of many new connections within the network as a whole. Increasing the order of the entire network from Order p to Order $p+1$ adds $(n-1)n^p$ new nodes, but requires only $2*(n-1)$ additional links to each existing node, ensuring that the overall number of connections grows linearly and slowly while the number of nodes grows exponentially. This satisfies our initial requirement for scalability.

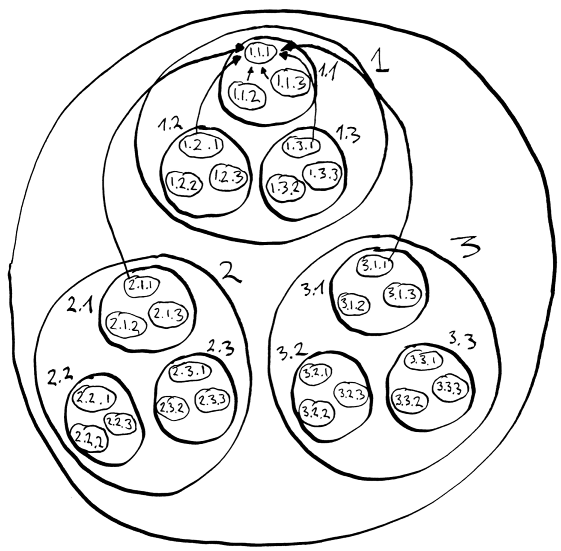

Figure 4: Third order network. Only incoming connections to Node 1.1.1 are shown.

A third order network is shown in Fig. 4, along with just those connections that are incoming to Node 1.1.1. While the network has grown by 18 nodes, Node 1.1.1 (formerly, Node 1.1) adds only 4 new connections, two outgoing and two incoming.

4.1 Network properties

Despite the wealth of reciprocal connections and recursive construction of the network, the two constraints on token passing outlined above ensure that every node receives tokens from every other node, but there is no duplication of tokens, and no cycles arise (connectivity). We show this in two steps. First, we show first that for any two non-identical nodes, A and B, there exists a unique path of transmission to B from A as constructed below. We then show that there is no transmission to B from A by any other path.

Path construction We begin at Node B. If there is a direct path of transmission from A to B, that is the shortest possible path. If there is no such direct path, we consider the cluster immediately containing A. Call this A'.

If there is a direct path of transmission from A' to B, then this is part of the unique path. Let the node transmitting the contents of A' be called C. The link from C to B is now part of the unique path. Iterate the procedure to find the path of transmission from A to C.

If there is no direct path of transmission from A' to B, consider A" (i.e. one cluster of higher order) and iterate. By construction there is a direct link from B to one node in the highest order cluster containing A and so this processes must terminate.

As an example, consider how audio from 1.1.1 is transmitted to 2.2.2. The first part of the path is from 2.2.2 to 1.2.2. Construction continues to identify the next segment, which is 1.2.2 to 1.1.2. Finally, the direct link from 1.1.2 to 1.1.1 is added.

No redundant transmission By Constraint 1, audio passing from a cluster of any order to another cluster of the same order contains only audio generated within the originating cluster. The procedure for finding the unique transmission path identified the connection of lowest order through which A's productions are transmitted to B, and this link is the only one at that level linking A and B. Links to B of lower order thus do not transmit anything from A.

Incoming links to B of higher order contain only audio generated outside the cluster containing both A and B (treating the entire network as the highest order cluster). By Constraint 2, they too do not transmit anything from A.

The tokens received by each node include tokens from every other node in the network, but each node receives them with its own unique set of delays. For example, in the second order network of Fig. 3, Node 2.1 receives tokens from Cluster 1 with lag 1 to Node 1.1.1 and lag 2 to Nodes 1.1.2 and 1.1.3. Node 2.2 likewise receives all tokens produced in Cluster 1, but with lag 1 to Node 1.2, and lag 2 to 1.1 and 1.3. This arrangement ensures that there is a unique stream of tokens at each node (singularity).

An interesting feature of the resulting network is its egalitarian structure. While Fig. 4 shows incoming links for Node 1.1.1, with which we started our network, an identical diagram can be drawn for every node in the network. The topology of a fully constructed network thus bears no trace of its history of construction, and every node participates in the same manner. We might call such a property 'perspectival homogeneity'.

With cluster size of 3, a network of the fifth order has 243 individual nodes. Each node has 10 incoming and 10 outgoing connections. Lags in transmission range from 1, between adjacent nodes, to a maximum of 5. In considering a large network like this, we may be interested in how close together two arbitrary nodes are. The manner in which the network was constructed makes neighbourhood relations among nodes somewhat non-obvious. Neighbourhood is here understood to be indexed by the lag in transmission from one node to another, i.e. it is defined in time, not in space. Nodes within a first order cluster are guaranteed to be near neighbours, that is, to exchange tokens with a lag of one. But near-neighbourhood for any given node extends to nodes in very distant clusters too. By way of illustration, Fig. 5 shows lags with respect to Node 1.1.1.1 in a fourth order network.

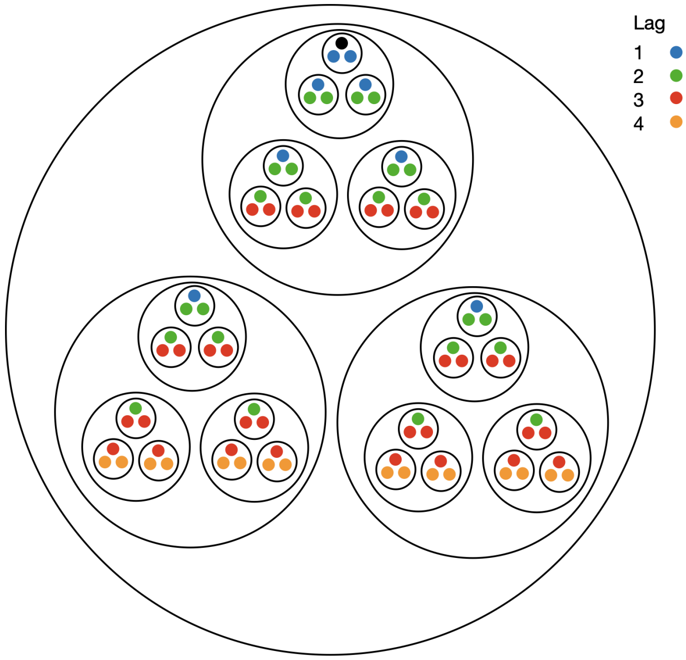

Figure 5: Lags shown with respect to Node 1.1.1.1 in a fourth order network.

5 Dynamics of a recurrently constructed network

We consider now a network in which the tokens exchanged are short utterances, such as a single repeated syllable "Bang!." We continue to finesse considerations of initialization and thus we assume a regular periodicity has been attained. The established period will act as a strong temporal constraint on all productions, and the absence of real-time reciprocal interaction means that slight temporal variations will not propagate easily through the network. However other forms of variation are possible. In fact, we can now distinguish clearly between temporal variability in production (largely eliminated) and all other forms of variation, such as pitch, intensity, etc. Local changes in any dimension other than time can propagate freely, such that if one node begins producing more intense tokens, this may be imitated by its immediate neighbours, and this variation may spread over time to remote nodes. For large networks, this opens the possibility of spatial patterning of variability that can spread conditioned by neighbourhood relations. For example, one producer may change to producing "Bing!" and this variation may be picked up and further propagated by near neighbours.

There arises now a space of possible experimentation in which large distributed networks of sound producers may synchronize their productions and collaboratively produce patterns of mutual influence, loosely analagous to how chant modulation propagates through large crowds in a football stadium. Empirical study of patterns of influence and co-variation can be conducted, for groups of participants with differing characteristics and collective goals.

The proposed topology and mode of operation have, as yet, been proposed abstractly, in order to ensure that actual implementation, which will inevitably be conditioned by contingencies of little relevance to the underlying idea, can rest upon a secure formal foundation. We plan to develop implementations of the proposed system using webaudio streams to pass tokens among nodes which will run in individual browsers or apps. Once networks can be constructed of non-trivial size and with participation of many widely distributed nodes, the problematic introduced at the start can be revisited with novel questions, and a rich space for experimental work.

6 From abstract clock to implementation

Finally we return to a problem we have conveniently omitted from the discussion so far—how to establish synchronization among the many nodes when there is, in fact, no God-clock driving node behaviour. Here, we restrict our consideration to first order clusters of nodes, i.e. to fully interconnected nodes with lag 1 between them.

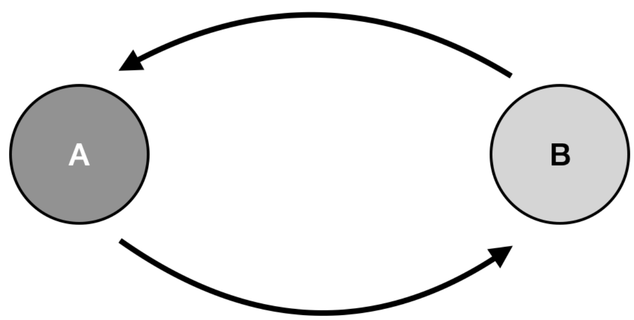

Figure 6: Minimal first order cluster of two nodes.

Consider first the smallest possible case of the dyad (Fig.). Let producer A initiate a roughly isochronous sequence of utterances of "Bang!" with period t. Under conventional transmission circumstances, B at first hears A's productions with some, perhaps slight, lag, and begins joining in. Before the known round-robin effect of cumulative lags becomes evident, we now interpose a monitoring process at Node B which receives tokens from A and *delays* release of the next A token until precisely where it estimates B's next production will lie. That is, a reset has occurred which, from B's point of view, ensures that A's tokens now appear to be well aligned with B's productions, at the cost of missing one production from A. A similar reset occurs at A, delaying B's tokens. Each participant now has the subjective impression that the two streams are well aligned. From the point of view of a hypothetical God clock, the two streams may be quite misaligned, but this is not evident to either participant.

At this point, we speculate on the stability of the situation, and find it lacking. Two participants linked in this fashion may continue to maintain isochronous production, but it is entirely possible that this is an unstable arrangement. Any temporal variation by one speaker will influence the other one cycle later, and it may look as if the round robin problem is now going to reappear, having only been deferred by one cycle. However we suggest that introduction of at least one other speaker will ensure that there is a collective stability in the arrangement that will lead to a very different outcome.

In Fig. 7 we have three fully interconnected nodes. Small temporal variation in the production of any one speaker now occurs in the context of the productions of two others, reducing the likelihood that it will directly elicit a compensatory response from anyone else. The addition of more speakers

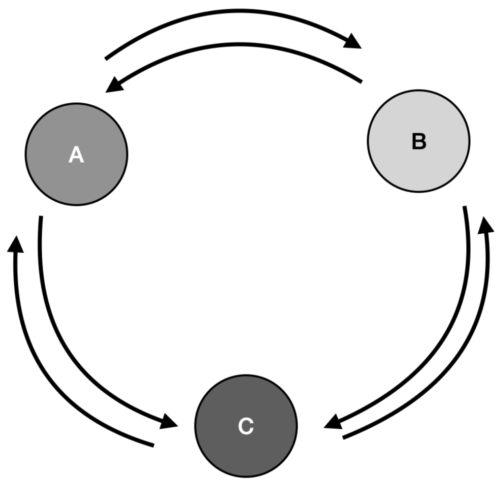

Figure 7: First order cluster of three nodes.

thus introduces a kind of temporal shock absorber, ensuring that there is a collective pressure to remain in time with the overall period of production. This buffering effect can be expected to be larger, as more nodes are added. Of course, this remains to be tested in implementation, but we believe that problems of instability become less severe as nodes are added, and the basline case of two nodes is not representative of the stability issues occurring in larger networks.

7 Discussion

We began with a motivation to seek a principled alternative in supporting distributed coordinated behaviours over networks that does not suffer from the logical problem of cumulative delays. Our inspiration was the desire to facilitate distributed chanting, but our treatment of the problem was, and remains, at a more abstract level which will need further work to bring it to implementation. Our chosen abstraction re-cast the problem from alignment in linear time to alignment in cyclic phase. With this conceptual shift in mind, we developed a recursively constructed network topology that exhibits our chosen properties of scalability to very many nodes, connectivity such that each node 'hears' every other node, singularity, i.e. the collective pattern heard by each node is distinguished by a unique set of lags betwen it and all other nodes, and perspectival homogeneity, such that no hierarchies or centers emerge. We have implemented the proposed network topology in simplified form where tokens passed are merely symbolic, and lags are enforced by an overall God clock. This implementation allowed us to verify that our desired criteria were met, and the topological properties of the networks are understood.

Building from here to a testable implementation requires several steps. As we have developed this model, we have looked to a future implementation based on webaudio streams. Current practical limitations on such streams undoubtedly exist, such as number of concurrent streams that can be simultaneously open for a single process, but such limitations can be expected to be overcome in the normal course of software advances. Initial testing of the stability properties of small clusters of fully-interconnected nodes could already be conducted.

As the space of possible implementations grows, we look forward to experimentation with the dynamic properties of such networks, examining how non-local patters of variation (temporal variation excluded) propagate, and what kind of neighbourhood effects can be seen. Because such experimentation will be exploratory, and may reveal unexpected properties, we remain blissfully unsure about the range of possible applications a network built along these lines might support. This, we believe, is a good place for basic research to set forth from.

8 Acknowledgements

This work grew from the generous support of the Digital Futures initiative that permitted a scholarly visit of the Fred Cummins to KTH in Summer, 2022.

References

Cummins, F. (2018). *The Ground From Which We Speak: Joint Speech and the Collective Subject.* Cambridge Scholars.