

# Distributed Implicit Timing Synchronization for Multihop Mesh Networks

Sriram Venkateswaran and Upamanyu Madhow

Department of ECE, University of California Santa Barbara, CA 93106

Email: {sriram, madhow}@ece.ucsb.edu

**Abstract**—Timing synchronization is a critical requirement for implementation of a number of performance-enhancing strategies in mesh networks, including Time Division Multiplexing (TDM) for more efficient resource usage, and sleep scheduling for power efficiency. While there is a significant body of prior work on distributed timing synchronization, it requires explicit exchange of timestamps; in addition, algorithms trying to synchronize clock frequencies as well as phases typically require exchange of clock rate estimates as well. In this paper, we investigate the feasibility of *overhead-free* maintenance of timing synchronization in a TDM-based mesh network, using only the information obtained from the timing of existing communication in the network: each node updates its clock rate and phase only when it receives a packet, depending on the difference between the expected and actual packet arrival times. While the algorithm is asynchronous (only a subset of nodes perform updates on any given slot), we exploit the law of large numbers to provide fundamental theoretical insight by analyzing an averaged, synchronous system. We demonstrate using simulations that a distributed, asynchronous algorithm designed based on prescriptions from the averaged system achieves phase and time synchrony.

## I. INTRODUCTION

Maintaining timing synchronization across a network of spatially dispersed nodes is important for a number of advanced functionalities, such as efficient resource sharing using Time Division Multiplexing (TDM), sleep scheduling for energy-efficient operation, and localization using time of arrival, or time difference of arrival, based techniques. In this paper, we investigate timing synchronization in the context of TDM-based resource sharing in a mesh network, and ask the following question: how well can timing synchronization be maintained without incurring any overhead related to timing-related signaling, using only the *implicit* timing information obtained from the TDM schedules themselves?

There are two key steps in network timing synchronization. At startup, different nodes are assumed to be completely asynchronous, so that the first step is to use explicit signaling to establish a common time reference. Such explicit signaling can also be employed to get estimates of the propagation delays (inclusive of processing delays) between the nodes. Since network set-up times of seconds or even minutes are acceptable, the one-time overhead of this procedure is not expected to be a bottleneck. We therefore focus in this paper on the second step of timing reference *maintenance*. Clocks at different nodes run at slightly different rates relative to a nominal rate due to manufacturing tolerances, which are typically of the order of 10-100 parts per million (ppm).

In addition, clock rates and phases can drift slowly over time due to temperature variations. Such phenomena causes the clock phases at different nodes to drift apart, and we must compensate for this drift “often enough” to maintain synchrony. One approach to timing reference maintenance is simply to rerun the first step of timing reference establishment at periodic intervals. However, this procedure incurs significant communication overhead. In this paper, we propose and investigate an adaptive approach to timing reference maintenance which exploits *implicit timestamps* from the TDM schedule to jointly adapt the clock phase and frequency at each node. When a node receives a packet from a neighbor, it compares the *actual* reception time with the *expected* reception time based on its own clock. The difference in these times are the inputs to its algorithm for clock phase and frequency adjustment. Thus, our approach is completely distributed, with each node adjusting its clock independently. Of course, these adjustments are all coupled through the network transmission schedule; the timing adjustments made by a node based on the timing of its received messages impact the times at which it transmits, and hence the adjustments made by nodes who receive these transmissions.

**Contributions:** Our key contributions are as follows:

- (a) We provide a completely distributed, implicit synchronization maintenance algorithm that compensates for both phase and frequency offsets, with frequency offset compensation occurring on a slower time scale. While our algorithm is asynchronous, with different subsets of nodes performing updates in different slots, we provide fundamental insight into feasibility and parameter choice by analyzing an averaged, synchronous system, in which nodes perform weighted updates, with weights depending on the transmission schedule in the original asynchronous system. The averaged system and the original system are linked through the law of large numbers (LLN), and it is shown via simulations that design prescriptions derived from the averaged system work for the original system.
- (b) For the averaged system, we prove that phase-only updates, which are attractive because of their simplicity, lead to irreducible phase errors: the magnitude of the worst-case pairwise phase offset depends on the distribution of clock rate skews in the network and grows with network size. We provide a linear programming formulation for finding the worst-case skews (from the point of maximizing the worst-case pairwise phase offset) for a given network topology. The phase offsets

for the averaged system are shown to be a lower bound for those for the original system. We conclude from our numerical results that phase-only updates may be acceptable in small networks, but do not scale to large networks.

(c) We then consider both phase and frequency updates, where, roughly speaking, the frequency updates are driven by the residual phase errors remaining after many phase updates. We prove phase and frequency convergence of our proposed scheme for the averaged system, and the conditions for convergence are used to infer the corresponding conditions on the transmission schedule for the original asynchronous system.

(d) We provide simulation results for the original asynchronous system showing that phase and frequency convergence are indeed attained when the algorithm parameters and the transmission schedule satisfy conditions derived from the averaged system.

Since our goal is to obtain fundamental insight into the feasibility of implicit timing synchronization, we do not model the physical and medium access layer of the underlying mesh network in detail. Our simulations are therefore based on the following abstractions: (a) the set of nodes that can successfully receive a packet sent by a given node, (b) the sets of *matchings*, or links that can be simultaneously active in a given TDM slot. We consider two scenarios: omnidirectional transmission, as is typical of current WiFi networks, and highly directional links, as in emerging millimeter (mm) wave networks [1]. The use of TDM is particularly attractive for the latter, since CSMA-based medium access is infeasible due to the deafness induced by highly directional links. The TDM schedules we consider for testing our timing synchronization method are chosen randomly (as described in Section VI) from a set of maximal matchings. We do not make any attempt to optimize or adapt the TDM schedules here, but the motivation for our timing synchronization scheme is to provide time slotting mechanisms with minimal guard periods, on top of which TDM-based medium access control (MAC) protocols can be built.

**Related Work:** While there is a vast literature on timing synchronization in networks, we believe this is the first paper to introduce the concept of *implicit* timestamps based on ongoing communication, to provide fundamental theoretical insight into attainable performance (via the averaged model), and to provide a completely decentralized algorithm for *both phase and frequency* adjustment. The well-known firefly-inspired algorithm [2] can be made to work with implicit timestamps, but it is only designed for phase synchrony, and does not handle either propagation delays or oscillator skew. The Reachback Firefly Algorithm (RFA) in [3] adapts the firefly algorithm to account for propagation delays in its phase adaptation, but it does not handle oscillator skew and therefore requires periodic resynchronization.

A number of algorithms ([4], [5], [6]) that have been proposed for distributed timing synchronization can be broadly classified as “consensus algorithms” [7]. These algorithms typically proceed in two stages: (1) estimate relative offset

and skew with each neighbor and (2) use neighbor’s estimates of offset and skew (typically with respect to a reference node) along with pairwise estimates to arrive at an estimate of one’s offset and skew. However, these algorithms are not amenable to an *implicit* implementation because all of them require *explicit* exchange of additional information to achieve consensus on *both frequency and phase*. The use of this information is critical for decoupling the problems of phase and frequency adjustments, in order to be able to run separate consensus algorithms for each of them. For example, [4] requires each node to broadcast the average error (both in rate and time) that it observes with its neighbors, while [5] and [6] require the nodes to broadcast their clock rates. The second order consensus algorithms proposed by [8] and [9] are not directly applicable because they also involve exchange of more information than timestamps (implicit or explicit); in order to use these algorithms for timing synchronization, the nodes would also need to exchange their clock rates. Reference [10] analyzes the algorithm proposed in [4] and shows that, for random connected networks, the error variance does not scale up with the size of network, thereby demonstrating the feasibility of using consensus-style algorithms for synchronization in large networks. In contrast, for our implicit model, phase-only adjustments lead to pairwise phase errors between neighbors which can increase with network size. To resolve this without requiring explicit messages, we need to introduce memory in our frequency adjustment rule, and run it on a slower time scale than the phase adjustment rule. While [11] also considers the use of memory, it does so to speed up convergence of *phase* synchronization, and does not consider clock skew.

Reference [12] characterizes the effect of asymmetric communication on consensus algorithms and identifies a scenario where the variance of the estimate can *increase* with more measurements because of lack of bidirectional communication. These are consistent with our own observations that convergence requires a “rich enough” communication pattern. The survey [13] provides a broad perspective on the field of distributed timing synchronization (see also the excellent discussion in [6]), while [14] identifies the fundamental limits of synchronizing “affine” clocks in the presence of delays, which includes but is not restricted to propagation delay. For distributed synchronization specifically applied to WiFi, see [15]. In contrast to the distributed algorithms discussed so far, references [16], [17] and [18] adopt a more centralized approach to timing synchronization. Reference [17] builds a rooted spanning tree on the network in the Level Discovery Phase and perform hop-by-hop synchronization, but it does not address clock skews, and therefore requires periodic resynchronization. Reference [18] uses linear regression to correct skew, with a dynamically elected root node flooding the network with its timestamp. As pointed out in [6], however, timing synchronization based on information flowing from a “distant” node, as in [18], performs poorly in certain topologies (specifically the ring topology). Reference [16] uses reference broadcasts from a transmitter to synchronize a cluster of receivers; while this is designed for single-hop

time synchronization, nodes that are common to two clusters can “convert” between the timestamps of two clusters.

**Outline:** In Section II, we describe the system model, including a description of our distributed algorithm for phase and frequency adjustment. We also introduce the averaged system that we use for developing analytical insight. In Section III, we analyze performance for the averaged system when we only adjust phase, and show how to compute the residual phase errors as a function of the frequency skews. This forms the building block for our design and analysis of phase and frequency adjustment rules in Section IV, where we prove convergence under a set of intuitively pleasing conditions for the averaged system. We then sketch the LLN-based link between the original system and the averaged system in Section V, and show how the results of Section IV are used to set design parameters for the original system. Simulations for the original system are provided in Section VI, and our conclusions are in Section VII.

## II. SYSTEM MODEL

We consider a network of  $N$  nodes, labeled  $\mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_N$ , each equipped with a clock that runs at a *nominal frequency* of  $f_{nom}$  Hz. The *actual* frequency of the clock at  $\mathcal{N}_i$ , denoted by  $f_i$ , differs slightly from the nominal rate of the clock  $f_{nom}$ , and is written as  $f_i = f_{nom}(1 + \rho_i)$  where  $\rho_i$  is called the skew, or drift, of the clock. The magnitude of the skew is assumed to be less than  $\rho_{max}$ , which is typically on the order of 10-100 parts per million (ppm); for example,  $\rho = 20 \times 10^{-6}$  corresponds to a skew of 20 ppm. Differences in the skews across nodes are caused by manufacturing tolerances. The variation of the skew at a given node with respect to time is typically very slow, hence we approximate  $\rho_i$  as a constant that takes values in  $[-\rho_{max}, \rho_{max}]$ .

The network employs a Time Division Multiplexed (TDM) schedule in which transmissions begin at integer multiples of a *slot time*, denoted by  $T_{slot}$ , according to the *transmitter’s clock*. It is convenient to describe the system from the point of view of an external observer who possesses a clock that runs at exactly the nominal rate  $f_{nom}$ . When the time on the external observer’s clock is  $t$ , let  $\varphi_i(t)$  denote the measure of time at  $\mathcal{N}_i$ . We refer to  $\varphi_i(t)$  as the clock *phase* of  $\mathcal{N}_i$ . Suppose that  $\mathcal{N}_j$  transmits to its neighbor  $\mathcal{N}_i$  in slot  $s$ . Therefore,  $\mathcal{N}_j$  starts transmitting when  $\varphi_j(t) = sT_{slot}$ . We assume that nodes have estimates of the propagation delays, including processing times, and subtract them out from the packet reception times. Thus,  $\mathcal{N}_i$  begins receiving this packet at  $\varphi_i(t)$ , and can implicitly conclude that its clock is behind  $\mathcal{N}_j$ ’s clock by  $\varphi_j(t) - \varphi_i(t) = sT_{slot} - \varphi_i(t)$ .  $\mathcal{N}_i$  can now reduce its phase error with respect to  $\mathcal{N}_j$  using the following linear phase adjustment.

**Linear Phase Adjustment:** Letting  $t^-$  and  $t^+$  denote the times on the external observer’s clock just before and after the phase jump, we have,

$$\varphi_i(t^+) = \varphi_i(t^-) + \beta[\varphi_j(t^-) - \varphi_i(t^-)] \quad (1)$$

where  $0 < \beta < 1$  is a design parameter.

**Quasi-Synchronous Approximation:** To simplify analysis, we assume that the phase and frequency adjustments are made at integer multiples of the slot times based on the external observer’s clock, rather than on the receiver’s clock (which keeps changing as we make phase and frequency adjustments). This approximation causes a second order error (because of measuring the phase offset between nodes at a time that is slightly offset from the true time) that is negligible compared to the phase offsets themselves. In simulations of the original system, we do not make this approximation, and verify that the quasi-synchronous approximation is indeed valid. We define  $\varphi_i[s] = \varphi_i(sT_{slot}^-)$  as the phase of the  $i$ th node just before the right edge of the  $s$ th slot. These phases are adjusted as in (1) at the slot boundaries, and then evolve linearly according to the relative frequency offsets of the nodes across a slot.

**Modeling skews:** The nodes change their frequencies only when they receive a packet and hence, skews can change explicitly only at slot boundaries. Let  $f_i[s]$  be the raw frequency of the  $i$ th node over the  $s$ th slot. Since we will primarily be dealing with the clock phases  $\varphi_i$ , it is convenient to normalize them and simplify the notation. To do this, we introduce an intermediate “raw phase” for node  $i$ , denoted by  $\theta_i(t)$ , which evolves across slot  $s$  as,

$$\theta_i((s+1)T_{slot}^-) = \theta_i(sT_{slot}^+) + f_i[s] \times T_{slot} \quad (2)$$

We define the clock phase  $\varphi_i(t)$  (as used above) to be  $\varphi_i(t) = \theta_i(t)/(f_{nom}T_{slot})$ . Let  $F_i[s] = \frac{f_i[s]}{f_{nom}}$  denote the corresponding normalized frequency. Define the *normalized mean frequency*

$$\bar{\psi}[s] = \frac{1}{N} \sum_{i=1}^N F_i[s]$$

This is the common part of the clock phase drift over the  $s$ th slot, and hence does not impact phase differences across nodes. The latter are actually driven by the normalized *excess frequencies*

$$\delta_i[s] = F_i[s] - \bar{\psi}[s], \quad i = 1, \dots, N$$

where  $\delta_i[s]$  is the amount by which the phase of the  $i$ th node drifts relative to the average.

**Discrete time dynamics:** When node  $i$  receives a packet from its neighbor node  $j$ , its phase evolves as follows:

$$\varphi_i[s+1] = \varphi_i[s] + \beta[\varphi_j[s] - \varphi_i[s]] + \bar{\psi}[s] + \delta_i[s] \quad (3)$$

According to the quasi-synchronous approximation, these phase updates occur synchronously at all nodes receiving packets, and can be conveniently expressed in vector form. Defining  $\varphi[s] = (\varphi_1[s], \varphi_2[s], \dots, \varphi_N[s])$  and  $\delta[s] = (\delta_1[s], \delta_2[s], \dots, \delta_N[s])$ , we have

$$\varphi[s+1] = G_s \varphi[s] + \delta[s] + \bar{\psi}[s] \mathbf{1} \quad (4)$$

where  $\mathbf{1}$  denotes the vector with all components equal to 1, and  $G_s$  is a matrix defined as follows: (1) If  $\mathcal{N}_i$  receives a packet from  $\mathcal{N}_j$  in slot  $s$ ,  $G_s(i, j) = \beta$ ,  $G_s(i, i) = 1 - \beta$  and  $G_s(i, j') = 0 \forall j' \neq i, j$  and (2) if  $\mathcal{N}_i$  does not

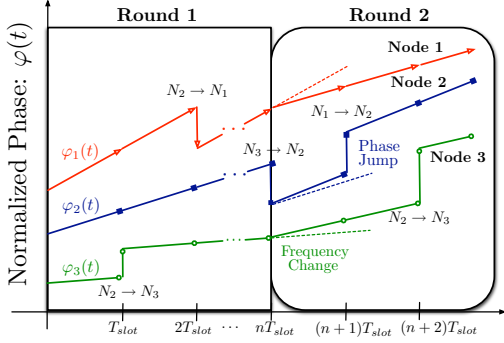


Fig. 1. Nodes make *phase jumps* each time they receive a packet. However, they change their frequencies (slope of the lines) only at the end of a *round* consisting of “many” slots.

receive a packet from any node in slot  $s$ ,  $G_s(i, i) = 1$  and  $G_s(i, j) = 0 \forall j \neq i$ . We refer to  $G_s$  as the *system matrix* at slot  $s$ . Note that  $G_s$  is a stochastic matrix (each row has nonnegative entries summing to one), so that  $G_s \mathbf{1} = \mathbf{1}$  (i.e.,  $\mathbf{1}$  is an eigenvector of  $G_s$  with eigenvalue 1).

**Frequency Adjustment:** We can see from (4) that excess frequencies lead to accumulation of phase errors between a node and its neighbors. Our frequency adjustment algorithm relies on this observation, and is loosely stated as follows (details provided later). A node concludes that its frequency is larger than the prevailing average in the network if its phase is ahead of its neighbors’ “on average” despite the phase adjustments it makes, where the averages are of phase errors measured over all neighbors over many slots. A representative diagram of the phase and frequency adjustments is sketched in Figure 1.

**Averaged System:** Each time  $\mathcal{N}_i$  receives a packet, it adjusts its clock phase based on the observed error with the transmitter. Since the node that transmits a packet to  $\mathcal{N}_i$  changes over slots, it introduces “fine details” into the dynamics of phase offset evolution that must be averaged out when estimating and compensating for skews. We therefore introduce the following fictitious *averaged system*: in each slot of the averaged system, every node updates its phase based on a weighted average of the phases of *all* its neighbors. The weights are derived from the communication pattern in the actual system and do not change with time. For example, for a random communication pattern, we model the matrices  $\{G_s\}_{s=0}^{\infty}$ , and thereby the set of links active concurrently, as being chosen independently and identically from a set  $\mathcal{S}_{\text{matching}} = \{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_M\}$  with probabilities  $\{p_1, p_2, \dots, p_M\}$  respectively. We denote the averaged system matrix by  $\bar{\mathcal{G}}$  and define it to be  $\bar{\mathcal{G}} = \sum_{i=1}^M p_i \mathcal{G}_i$ . The phase evolution in the averaged system is therefore exactly as in (4), except that  $G_s$  is replaced by  $\bar{\mathcal{G}}$ .

$$\varphi[s+1] = \bar{\mathcal{G}}\varphi[s] + \delta[s] + \bar{\psi}[s]\mathbf{1} \quad (5)$$

For simplicity, we assume throughout this paper that the averaged system matrix  $\bar{\mathcal{G}}$  is symmetric. This corresponds to the links in the network having the same probability of being active in either direction.

**Network Start-up:** While our focus here is developing a theory of implicit synchronization *maintenance*, our simulations do include a gateway-led start-up scheme for coarse initial synchronization (no attempt is made to optimize this). A gateway node broadcasts its time and other nodes in its neighborhood “set” their clocks to this value. Each of these nodes then broadcasts its time enabling nodes that are two hops from the gateway to set their clocks. This broadcast is made at a random instant in an interval  $[\tau_{\min}, \tau_{\max}]$  after its clock was set. If  $\mathcal{N}_i$ ’s clock is already set when it hears  $\mathcal{N}_j$ ’s broadcast, it simply adjusts its clock to the mean value. Specifically, if  $\mathcal{N}_j$ ’s time when it makes the broadcast is  $\varphi_j$  and at this point,  $\mathcal{N}_i$ ’s time is  $\varphi_i$ ,  $\mathcal{N}_i$  adjusts its clock to  $(\varphi_j + \varphi_i)/2$ . Each node makes one such broadcast and the network can be coarsely synchronized in this fashion in a time period that scales linearly with the number of nodes. We assume throughout the paper that nodes have good estimates of the propagation delay to their neighbors. This can be done by exchanging multiple packets with neighbors at the time of network startup and estimating the propagation delay in a manner similar to [14]. Any residual errors in estimating the propagation delay are considered to be subsumed in the phase jitter in our model.

### III. PHASE-ONLY ADJUSTMENTS

In this section, we analyze the performance of a system in which the nodes adjust only their clock phases, and never their frequencies. Phase-only updates are of interest in their own right (due to their simplicity), while also being the first step in the design of phase-frequency updates. The uncompensated skews in such a system tend to drive the network towards asynchrony, while the recurrent phase compensations tend to pull the phases closer to each other. Our metric for quantifying which of these two forces is dominant is the worst-case pairwise clock phase error between neighbors, which is precisely the overhead required to maintain a TDM schedule. We first analyze this metric for the averaged system, and then observe that this provides a lower bound for the original system.

#### A. Averaged System

Starting with (5), note that the mean and excess frequencies,  $\bar{\psi}[s]$  and  $\delta[s]$ , do not change with  $s$  when we only adjust phases. We can therefore drop the time index  $s$  and denote these quantities by  $\bar{\psi}$  and  $\delta$ , respectively. Iterating, the phases in slot  $s$  can be expressed in terms of the initial conditions as

$$\varphi[s] = \bar{\mathcal{G}}^s \varphi[0] + s\bar{\psi}\mathbf{1} + (\mathbb{I} + \bar{\mathcal{G}} + \bar{\mathcal{G}}^2 + \dots + \bar{\mathcal{G}}^{s-1})\delta \quad s \geq 1 \quad (6)$$

As with the frequencies, it is useful to decompose the phases into mean and excess phases, as follows:

$$\varphi[s] = \bar{\varphi}[s]\mathbf{1} + \varphi_{ex}[s]$$

where  $\bar{\varphi}[s] = \mathbf{1}^T \varphi[s]/N = \frac{1}{N} \sum_{i=1}^N \varphi_i[s]$  is the average phase and  $\varphi_{ex}[s]$  represents the excess phase.

We begin with an eigendecomposition of  $\bar{\mathcal{G}}$  as,

$$\bar{\mathcal{G}} = \sum_{l=1}^N \lambda_l \mathbf{v}_l \mathbf{v}_l^T \quad (7)$$

where (a)  $\bar{\mathcal{G}} \mathbf{v}_l = \lambda_l \mathbf{v}_l$ , (b) the eigenvalues are arranged in descending order of magnitude  $|\lambda_1| \geq |\lambda_2|, \dots, \geq |\lambda_N|$ , (c) the eigenvectors  $\mathbf{v}_l$  have unit norm, or  $\mathbf{v}_l^T \mathbf{v}_l = 1$  and (d) eigenvectors corresponding to different eigenvalues are orthogonal, or  $\mathbf{v}_i^T \mathbf{v}_j = 0$   $i \neq j$ , because  $\bar{\mathcal{G}}$  is symmetric. Because  $\bar{\mathcal{G}}$  is stochastic, its largest eigenvalue is one, with  $\mathbf{v}_1 = \mathbf{1}/\sqrt{N}$  being the corresponding unit norm eigenvector. For convergence of consensus style algorithms, this largest eigenvalue must not be repeated. The condition for this is that the graph corresponding to  $\bar{\mathcal{G}}$  must be connected [19] (clearly needed for network-wide convergence). In this case, we can approximate  $\bar{\mathcal{G}}^s$  by the dominant term in its spectral decomposition for large values of  $s$  as,

$$\bar{\mathcal{G}}^s \approx \lambda_1^s \mathbf{v}_1 \mathbf{v}_1^T = \frac{1}{N} \mathbf{1} \mathbf{1}^T \quad (8)$$

for reasonably large  $s$ , so that the first term in (6) tends to  $\mathbf{1}^T \varphi[0]/N$ , the mean phase at start-up. The second term in (6) also contributes only to the mean phase. For the third term, the eigenmode corresponding to eigenvalue one has no response, since the excess frequency  $\delta$  has zero mean. Thus, the evolution of the excess phase (which is what drives our phase-only algorithm) can be obtained by excising the first eigenmode from  $\bar{\mathcal{G}}$  to get the matrix

$$\bar{\mathcal{G}}_{ex} = \bar{\mathcal{G}} - \frac{\mathbf{1} \mathbf{1}^T}{N} = \sum_{l=2}^N \lambda_l \mathbf{v}_l \mathbf{v}_l^T \quad (9)$$

which contains only eigenmodes with eigenvalues with magnitude strictly smaller than one (the dynamics induced by this matrix are dominated by the magnitude of the second largest eigenvalue,  $|\lambda_2|$ ). For large  $s$ , the excess phase therefore exhibits the following behavior:

$$\begin{aligned} \varphi_{ex}[s] &= \left( \mathbb{I} + \bar{\mathcal{G}}_{ex} + \bar{\mathcal{G}}_{ex}^2 + \dots + \bar{\mathcal{G}}_{ex}^{s-1} \right) \delta \\ &\rightarrow \left( \mathbb{I} - \bar{\mathcal{G}}_{ex} \right)^{-1} \delta, \quad s \rightarrow \infty \end{aligned} \quad (10)$$

We provide a detailed derivation of this result in Section VIII-A. Thus, the excess phases exhibit an irreducible error floor that depends on the skews. To quantify the impact of this, we maximize the worst-case pairwise phase errors between neighboring nodes over the set of possible skews. We show, in section VIII-B, that the pairwise error between a fixed pair of neighbors can be maximized by solving a linear programming (LP) problem, where the constraints are determined by maximum allowable skew  $\rho_{max}$ . We then maximize over the solutions of the LPs corresponding to all possible directed edges to get the worst-case pairwise phase error. Since all of these LPs share the same feasible set, and the optimal solution for any LP occurs at an extreme point [21], we show, by characterizing the extreme points of the feasible set, that the worst phase error between neighboring nodes is produced by (roughly) half the node clocks running at the

largest frequency and the other half running at the smallest frequency for *any* network topology. A detailed derivation of these facts is provided in Section VIII-B.

### B. Actual System

We now show that the averaged system provides a bound on the errors between neighbors in the actual system. We sketch the central ideas here and provide the details in Section VIII-C. Iterating (4), we obtain that the phase of the actual system evolves as follows:

$$\begin{aligned} \varphi[s] &= G_{s-1} G_{s-2} \dots G_0 \varphi[0] + s \bar{\psi} \mathbf{1} \\ &\quad + \left( \mathbb{I} + \sum_{k=1}^{s-1} G_{s-1} G_{s-2} \dots G_{s-k} \right) \delta \quad s \geq 2 \end{aligned} \quad (11)$$

The second term only contributes to the mean phase. Under a suitable connectedness condition [19] which is needed for consensus, we can show that the same is true for the first term as well as  $s$  gets large:

$$\lim_{s \rightarrow \infty} G_s G_{s-1} \dots G_0 = \mathbf{1} \gamma^T \quad (12)$$

where the elements of  $\gamma$  are nonnegative and  $\gamma^T \mathbf{1} = 1$ . We can therefore throw out the first and second terms in (11) when considering the evolution of the excess phases, and obtain

$$\varphi_{ex}[s] \approx \left( \mathbb{I} + \sum_{k=1}^{s-1} G_{s-1} G_{s-2} \dots G_{s-k} \right) \delta \quad (13)$$

Let us now average the evolution over many realizations of the communication schedule. For each schedule, the matrices  $G_n$  are chosen in i.i.d. fashion from the set  $\mathcal{S}_{matching} = \{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_M\}$  with probabilities  $\{p_1, p_2, \dots, p_M\}$  and the average across this choices is denoted by  $\bar{\mathcal{G}}$ , the matrix for our averaged system (the corresponding excised matrix is  $\bar{\mathcal{G}}_{ex}$ ). Using the LLN, the averaged trajectory of excess phases satisfies  $\bar{\varphi}_{ex}[s] \approx \left( \mathbb{I} + \sum_{k=1}^{s-1} \bar{\mathcal{G}}^k \right) \delta$ . Since  $\bar{\mathcal{G}}$  is symmetric, we can excise its first eigenmode to obtain,

$$\bar{\varphi}_{ex}[s] \approx \left( \mathbb{I} + \sum_{k=1}^{s-1} \bar{\mathcal{G}}_{ex}^k \right) \delta \quad (14)$$

which is the equation we had for the phase evolution in the averaged system. The maximum value of the pairwise difference between neighbors' phases (or equivalently, excess phases) is given by  $\|\mathbf{C} \varphi_{ex}[s]\|_\infty$ , where matrix multiplication by  $\mathbf{C}$  corresponds to all possible pairwise differences across neighbors and  $\|\mathbf{x}\|_\infty$  denotes the infinity norm, or maximum element, of a vector  $\mathbf{x}$ . This is a convex function of  $\varphi$ . We can now apply Jensen's inequality to infer that the maximum pairwise phase difference for the averaged system is a lower bound on the average of the maximum pairwise phase difference in the original system, averaged across realizations. Consequently, any set of skews that are "bad" for the averaged system are guaranteed to be bad for the actual system as well. We use this to show in Section VI that the mean value of the worst-case pairwise error between neighbors can increase with the size of the network, so that both frequency and phase adjustments are required for large networks.

#### IV. DESIGN OF PHASE-FREQUENCY ADJUSTMENTS

We now present an algorithm that achieves frequency and phase synchrony for the averaged system. Equation (10) shows that, after sufficiently many phase-only adjustments, the excess phases, and hence the residual phase differences between nodes, are a function of the skews. Conversely, these residual phase differences provide estimates of the skews, which are then used to make frequency adjustments. We note from (10) that  $\delta \approx (\mathbb{I} - \overline{\mathcal{G}}_{ex}) \varphi_{ex}[s]$  for large enough  $s$ . For decentralized adaptation, we clearly do not have access to the excess phases  $\varphi_{ex}$ . However, using the fact that  $\overline{\mathcal{G}}$  is a stochastic matrix, we can also express the skews in terms of the raw phases as follows (see Section VIII-D for details) :

$$\delta \approx (\mathbb{I} - \overline{\mathcal{G}}) \varphi[s] = \overline{\mathcal{L}}\varphi[s] \quad (15)$$

for sufficiently large  $s$ , where the matrix  $\overline{\mathcal{L}} = \mathbb{I} - \overline{\mathcal{G}}$  is called the Laplacian of the system. For our phase-only updates, it is easy to show (see Section VIII-E) that the  $i$ th component of (15) can be written as

$$[\overline{\mathcal{L}}\varphi_\infty]_i = \beta \sum_{j=1}^N q_{j \rightarrow i} (\varphi_{\infty,i} - \varphi_{\infty,j}) = \delta_i \quad (16)$$

where  $q_{j \rightarrow i}$  denotes the probability that  $\mathcal{N}_j$  transmits to  $\mathcal{N}_i$  (even though the summation is defined over all the nodes, we note that the only terms that contribute are those with  $q_{j \rightarrow i} > 0$  i.e. nodes that are neighbors of  $\mathcal{N}_i$ ). Thus, the  $i$ th node can estimate its excess frequency simply by considering the weighted average of its residual phase differences with its neighbors, with weights depending on the average communication pattern defined by  $\overline{\mathcal{G}}$ .

Thus, our phase-frequency adjustment algorithm consists of estimating the skews based on the residual phase errors with neighbors after many phase-only adjustments, and adjusting the frequencies accordingly. We follow multiple rounds of this procedure, starting from round 0, with round  $r$  consisting of  $W_r$  slots. Each node employs the same algorithm, hence we describe it from the point of view of a particular node  $i$ :

- *Step 1:* In round  $r$ , node  $i$  makes phase-only adjustments for  $W_r$  slots;
- *Step 2:* At the end of round  $r$ , node  $i$  makes an estimate of its current skew, denoted by  $\hat{\delta}_i$ , by averaging the observed errors with its neighbors in the following slot according to (16);
- *Step 3:* If the skew is small enough, with  $|\hat{\delta}_i| < \epsilon$  falling in a “dead zone,” node  $i$  does not change its frequency. If  $|\hat{\delta}_i| > \epsilon$ , then node  $i$  makes a frequency jump of  $\pm\mu$ , depending on the sign of the skew. That is, it adds  $\Delta F_i = -\mu \text{sign}(\hat{\delta}_i)$  to its frequency. (The dead zone parameter  $\epsilon$  and the step size parameter  $\mu$  are discussed below.)
- *Step 4:* Go back to Step 1 for round  $r + 1$ .

**Algorithm parameters:** We prove that our frequency adjustment algorithm guarantees that the maximum frequency deviation away from the mean of any node is at most  $\epsilon + \mu + \chi$ , where  $\epsilon$  is the dead zone size,  $\mu$  is the frequency adjustment step size, and  $\chi$  is the designed error in frequency estimation

at the end of each round. The parameters  $\epsilon, \mu, \chi > 0$  can be chosen freely, except that our convergence proof requires that  $\epsilon > \mu + \chi$  (the dead zone must be large enough to accommodate wrong frequency steps and frequency estimation errors). For example, starting with a maximum frequency offset of 100 ppm, we could eventually converge to a maximum offset of less than 4 ppm by choosing  $\epsilon = 2$  ppm,  $\mu = 1$  ppm and  $\chi = 0.8$  ppm. This would reduce the worst-case phase offset in the network by a factor of 25. The round sizes  $\{W_r\}$  are chosen such that the maximum error in skew estimation at the end of each round is  $\chi$ .

Our theoretical development proceeds as follows. We first identify the number of slots needed in round  $r$ , which we denote by  $W_r$ , so that the maximum error in estimating the current skew of any node is smaller than  $\chi$ . We then show that, if  $\epsilon > \mu + \chi$ , then in each round of frequency adjustment, either (a) the nodes move towards convergence or (b) all current skews are lesser than  $\epsilon + \mu + \chi$  (in magnitude) and will continue to remain so in the future. It is convenient at this point to introduce two-dimensional indexing of slots: slot  $[s, r]$  is the  $s$ th slot in round  $r$ , and is therefore the  $k$ th slot in a one-dimensional enumeration of slots, where  $k = \sum_{q=0}^{r-1} W_q + s$ . In the  $r$ th round, the slot index  $s$  takes values between 0 and  $W_r - 1$ . We use both one-dimensional and two-dimensional enumeration of slots, depending on the context.

##### A. Choosing round sizes

Since the mean phase does not contribute to the actions of our algorithms (which depend only on phase *differences*), we only need to model what happens to the excess phases. As in Section III-A, this means that we can excise the eigenmode corresponding to eigenvalue one, which determines evolution of the mean phase. We show in Section VIII-F that the excess phase in slot  $s$  of round  $r$  evolves as

$$\varphi_{ex}[s, r] = \overline{\mathcal{G}}_{ex}^s \varphi_{ex}[0, r] + D_s \delta[0, r] \quad (17)$$

where  $D_s = \mathbb{I} + \overline{\mathcal{G}}_{ex} + \overline{\mathcal{G}}_{ex}^2 + \dots + \overline{\mathcal{G}}_{ex}^{s-1}$ . Assuming that slot  $s = W_r - 1$  is the last one in round  $r$  (i.e., the nodes estimate their current skews as  $\overline{\mathcal{L}}\varphi[s, r]$ ), the current skew estimates at the end of round  $r$  are given by

$$\hat{\delta}[s, r] = (\mathbb{I} - \overline{\mathcal{G}}) \varphi[s, r] = (\mathbb{I} - \overline{\mathcal{G}}_{ex}) \varphi_{ex}[s, r] \quad (18)$$

Substituting the expression for the excess phases from equation (17) into equation (18), we get an expression for the errors incurred by the nodes with this estimate. Denoting these errors by  $e_\delta[s, r] = \hat{\delta}[s, r] - \delta[0, r]$ , we get

$$e_\delta[s, r] = (\overline{\mathcal{G}}_{ex}^s - \overline{\mathcal{G}}_{ex}^{s+1}) \varphi_{ex}[0, r] - \overline{\mathcal{G}}_{ex}^s \delta[0, r] \quad (19)$$

We approximate  $\overline{\mathcal{G}}_{ex}^s$  by its dominant eigenmode  $\lambda_2^s \mathbf{v}_2 \mathbf{v}_2^T$  and thereby, the errors as,

$$e_\delta[s, r] \approx \lambda_2^s \mathbf{v}_2 \mathbf{v}_2^T (1 - \lambda_2) \varphi_{ex}[0, r] - \lambda_2^s \mathbf{v}_2 \mathbf{v}_2^T \delta[0, r] \quad (20)$$

We can now “invert” this relationship to show that the estimation error can be made arbitrarily small by waiting “long enough”. Specifically, the maximum error in estimating the

excess frequency in round  $r$  can be made smaller than  $\chi$  by choosing the number of slots  $W_r$ , to satisfy,

$$W_r \geq \frac{1}{\log(1/\lambda_2)} \times \left( \log \frac{\chi}{\|\mathbf{v}_2 \mathbf{v}_2^T\|_\infty} - \log \{ \|\varphi_{ex}[0, r]\|_\infty + \|\delta[0, r]\|_\infty (1 - \lambda_2) \} \right) \quad (21)$$

Next, we explain how the length of the rounds  $W_r$  can be computed in a recursive fashion. For the first round, the coarse synchronization procedure at startup provides bounds on the excess phase  $\|\varphi_{ex}[0, 1]\|_\infty$ . Since the node frequencies are bounded at the time of startup, we can show that  $\|\delta[0, 1]\|_\infty$  is bounded by  $2\rho_{max}$ . These bounds are used in equation (21) to calculate the number of slots in the first round  $W_1$ . In Section IV-B, we show that the maximum frequency in round 2 does not increase from its value in round 1, despite the frequency adjustments made by nodes at the end of round 1. Therefore,  $\|\delta[0, 2]\|_\infty$  is also bounded by  $2\rho_{max}$ . The excess phases are bounded in a recursive fashion: given bounds on  $\|\varphi_{ex}[0, r]\|_\infty$  and  $\|\delta[0, r]\|_\infty$ , we obtain a bound on  $\|\varphi_{ex}[0, r+1]\|_\infty$ . Specifically, the excess phase at the start of round 2 can be bounded by setting  $r = 1$  in the recursive relationship (shown in Section VIII-G),

$$\|\varphi_{ex}[0, r+1]\|_\infty \leq \|\overline{\mathcal{G}}_{ex}^{W_r-1}\|_\infty \|\varphi_{ex}[0, r]\|_\infty + (1 + \|D_{W_r-1}\|_\infty) \|\delta[0, r]\|_\infty \quad (22)$$

Note that this bound depends explicitly on  $W_1$ , which we have already calculated. Substituting these bounds for  $\delta[0, 2]$  and  $\varphi_{ex}[0, 2]$  in equation (21), we can calculate the number of slots needed in round 2. By repeating these arguments, we can recursively compute the number of slots needed in any round  $r$ .

### B. Convergence of frequency adjustment algorithm

By making each round long enough, the errors in estimating the excess frequencies can be made smaller than  $\chi$  where  $\chi$  can be arbitrarily small. We demonstrate that each frequency adjustment drives the network towards (and never away from) frequency synchrony. For this, we need to choose the width of the “dead zone”  $\epsilon$  to be larger than the sum of the estimation error  $\chi$  and the frequency adjustment size  $\mu$ . Specifically, we show that, after “many” rounds of frequency adjustments, the excess frequencies are no bigger than  $\epsilon + \mu + \chi$ . Since the latter quantity can be made as small as we wish, we can get arbitrarily close to perfect frequency synchrony.

Since frequencies change only over rounds, we index the current skews by round number: the current skew of node  $i$  in any slot of the  $r^{th}$  round is denoted by  $\delta_i\{r\}$ . We do not need to track phases any more, but rather repeatedly use the fact that the error in estimating the excess frequency is smaller than  $\chi$ , or  $|\delta_i\{r\} - \hat{\delta}_i\{r\}| \leq \chi \forall i, r$ .

Consider a given round  $r$ . We show that nodes with current skews of large magnitude are guaranteed to change their frequency in the right direction, and that nodes never move their frequency in the wrong direction. To this end, we split the

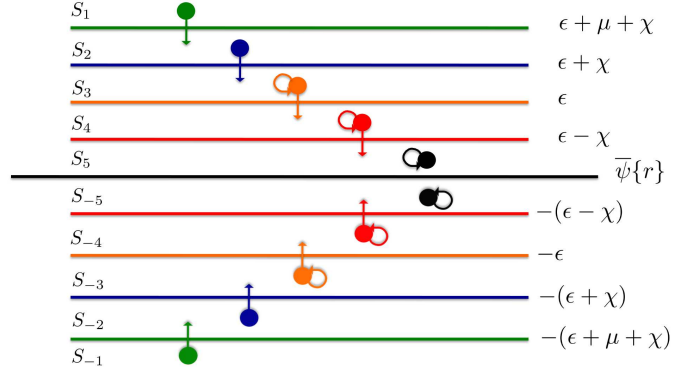


Fig. 2. Splitting the nodes into 10 sets based on their excess frequencies. Nodes in  $S_1$  and  $S_2$ , that are “far” from convergence, are guaranteed to reduce their frequencies. Nodes in  $S_3$  and  $S_4$ , that are “close” to convergence, either reduce their frequencies or do not change it, but never increase their frequencies. Nodes in  $S_5$ , that are “closest” to convergence, will not change their frequencies. Analogous results hold for nodes with negative excess frequencies.

nodes into ten sets based on the sign of their excess frequencies and how far they are from convergence. The split is shown in Figure 2. The first set  $S_1\{r\}$  contains nodes  $\delta_i\{r\} > \epsilon + \mu + \chi$  (positive skews, “very far” from convergence). The second set  $S_2\{r\}$  contains nodes with  $\epsilon + \chi < \delta_i\{r\} \leq \epsilon + \mu + \chi$  (positive skews, “far from convergence”). The next three sets  $S_3\{r\}, S_4\{r\}, S_5\{r\}$  contain nodes that are “close”, “closer” and “closest” to convergence: for  $S_3\{r\}$ ,  $\delta_i\{r\} \in (\epsilon, \epsilon + \chi]$ ; for  $S_4\{r\}$ ,  $\delta_i\{r\} \in (\epsilon - \chi, \epsilon]$ ; for  $S_5\{r\}$ ,  $\delta_i\{r\} \in (0, \epsilon - \chi]$ . The sets  $S_{-i}$  are defined analogously, except that the signs are reversed, as shown in the figure. We now prove a series of propositions that characterize the actions of nodes in each of these sets. We then put these propositions together to show that the algorithm converges and achieves frequency synchrony (to within  $\epsilon + \mu + \chi$ ).

First, we show that a node with a positive current skew in round  $r$  either lowers its frequency by  $\mu$  or does not change it at all. It never increases its frequency further by  $\mu$ . Furthermore, we show that nodes which are either far or very far from convergence are guaranteed to reduce their frequencies by  $\mu$ .

**Proposition 1.** *If  $F_i\{r\} \geq \overline{\psi}\{r\}$ , then  $F_i\{r+1\} \leq F_i\{r\}$ . Furthermore, if  $N_i \in S_1\{r\} \cup S_2\{r\}$ , then  $F_i\{r+1\} = F_i\{r\} - \mu$ . Analogously, if  $F_i\{r\} \leq \overline{\psi}\{r\}$ , then  $F_i\{r+1\} \geq F_i\{r\}$ , and if  $N_i \in S_{-1}\{r\}/S_{-2}\{r\}$ , then  $F_i\{r+1\} = F_i\{r\} + \mu$ .*

*Proof:* We only prove the statement for nodes with positive current skews, or  $F_i\{r\} \geq \overline{\psi}\{r\}$ , since the proof for nodes with  $F_i\{r\} \leq \overline{\psi}\{r\}$  is entirely analogous.

*Case 1:* If node  $i$  is in  $S_1\{r\} \cup S_2\{r\}$ , then  $\delta_i\{r\} > \epsilon + \chi$ . Since the estimation error is bounded by  $\chi$  (we have ensured this by making the rounds long enough), the estimated current skew falls outside the dead zone:  $\hat{\delta}_i\{r\} \geq \delta_i\{r\} - \chi > (\epsilon + \chi) - \chi = \epsilon$ . Thus, node  $i$  adjusts its frequency downwards.

*Case 2:* For nodes in  $S_3, S_4$  or  $S_5$ , the current skew is positive:  $0 < \delta_i\{r\} < \epsilon + \chi$ . Since the estimation error is at most  $\chi$ , we have  $-\chi < \hat{\delta}_i\{r\} < \epsilon + 2\chi$ . Since  $\epsilon > \mu + \chi > \chi$ , we conclude that  $\hat{\delta}_i\{r\} > -\epsilon$ , so that node  $i$  does not increase its frequency. ■

Next, we show that the frequency jumps are small (relative to the size of the dead zone), so that nodes do not cross one another while making frequency adjustments. Specifically, a node with a positive excess frequency in round  $r$  that makes a downward frequency jump never crosses a node with negative excess frequency in round  $r$  that is increasing its frequency.

**Proposition 2.** *If  $F_i\{r\} > \bar{\psi}\{r\}, F_i\{r+1\} > \bar{\psi}\{r\}$ . Analogously, if  $F_i\{r\} < \bar{\psi}\{r\}, F_i\{r+1\} < \bar{\psi}\{r\}$ .*

*Proof:* Once again, we only prove the statement for nodes with frequencies larger than the mean, since the other case is exactly analogous.

*Case 1:* Let  $\mathcal{N}_i \in S_5\{r\}$ . Therefore,  $0 \leq \delta_i\{r\} \leq \epsilon - \chi$ . Since the estimation error is bounded by  $\chi$ , we have  $-\chi \leq \hat{\delta}_i\{r\} \leq \epsilon$ . Since  $\epsilon > \chi + \mu$ , we have  $-\chi > -\epsilon$ . Therefore, the estimate  $\hat{\delta}_i\{r\}$  falls in the dead zone and  $\mathcal{N}_i$  does not change its frequency at the end of round  $r$ . Thus, we have  $F_i\{r+1\} = F_i\{r\}$  and by definition,  $F_i\{r\} > \bar{\psi}\{r\}$ , which together imply  $F_i\{r+1\} > \bar{\psi}\{r\}$ .

*Case 2:* If  $\mathcal{N}_i$  is in  $S_1\{r\} \cup S_2\{r\} \cup S_3 \cup \{r\} \cup S_4\{r\}$ , we have  $F_i\{r\} > \bar{\psi}\{r\} + \epsilon - \chi$ . At the end of round  $r$ ,  $\mathcal{N}_i$  can reduce its frequency at most by  $\mu$ . Therefore,  $F_i\{r+1\} \geq F_i\{r\} - \mu > \bar{\psi}\{r\} + \epsilon - \chi - \mu$ . Since  $\epsilon > \chi + \mu$ ,  $F_i\{r+1\} > \bar{\psi}\{r\}$ . ■

We denote the maximum and minimum frequencies across nodes in round  $r$  by  $F_{max}\{r\}$  and  $F_{min}\{r\}$  respectively. We show that the maximum frequency  $F_{max}\{r\}$  can never increase from round  $r$  to round  $r+1$ . It can only decrease or remain unchanged. A similar result holds for the minimum frequency as well.

**Proposition 3.** *The maximum frequency can never increase and the minimum frequency can never decrease i.e.  $F_{max}\{r+1\} \leq F_{max}\{r\}$  and  $F_{min}\{r+1\} \geq F_{min}\{r\}$ .*

*Proof:* We only prove that  $F_{max}$  can never increase, since the proof that  $F_{min}$  can never decrease is similar. Let  $\Gamma^+\{r\}$  and  $\Gamma^-\{r\}$  denote the set of all nodes with positive and negative excess frequencies in round  $r$  respectively i.e.  $\Gamma^+\{r\} = \cup_{i=1}^5 S_i\{r\}$  and  $\Gamma^-\{r\} = \cup_{i=1}^5 S_{-i}\{r\}$ . Unless all nodes are at the same frequency (in which case we are done), neither of these sets can be empty. Otherwise, all the nodes would have their frequencies on one side of the mean frequency, which is impossible.

Let  $u^+$  and  $u^-$  be any nodes in  $\Gamma^+\{r\}$  and  $\Gamma^-\{r\}$  respectively. From Proposition (2), we have  $F_{u^+}\{r+1\} > \bar{\psi}\{r\}$  and  $F_{u^-}\{r+1\} < \bar{\psi}\{r\}$ . Therefore, we have  $F_{u^+}\{r+1\} > F_{u^-}\{r+1\} \forall u^+, u^-$ . Thus, the node with the maximum frequency in round  $r+1$ , denoted by  $u_{r+1}^*$ , must have had a positive excess frequency in round  $r$ , or  $u_{r+1}^* \in \Gamma^+\{r\}$ . But, if  $u_{r+1}^* \in \Gamma^+\{r\}$ , by Proposition 1, its frequency in round  $r+1$  cannot be larger than its frequency in round  $r$  i.e.  $F_{u_{r+1}^*}\{r+1\} \leq F_{u_{r+1}^*}\{r\}$ . Since the maximum frequency

across nodes in round  $r$  is larger than  $u_{r+1}^*$ 's frequency in the same round, we get  $F_{u_{r+1}^*}\{r\} \leq F_{max}\{r\}$ . Chaining these inequalities, we get,  $F_{max}\{r+1\} = F_{u_{r+1}^*}\{r+1\} \leq F_{u_{r+1}^*}\{r\} \leq F_{max}\{r\}$ . Therefore, the maximum frequency can never increase across rounds. ■

We also get an intuitively pleasing conclusion from this proposition: if we achieve frequency synchrony, the common frequency will be between the maximum and minimum frequencies at the start. We now go one step further and show that if some nodes are “very far” from convergence, the maximum frequency will decrease. An analogous result holds for the minimum frequency too.

**Proposition 4.** *If  $S_1\{r\} \neq \emptyset, F_{max}\{r+1\} = F_{max}\{r\} - \mu$ . Furthermore, if  $\mathcal{N}^\#$  is the node with the maximum frequency in round  $r$ , it continues to be the node with the maximum frequency in round  $r+1$ . Analogously, if  $S_{-1}\{r\} \neq \emptyset, F_{min}\{r+1\} = F_{min}\{r\} + \mu$ . Also, if  $\mathcal{N}^\#$  is the node with the minimum frequency in round  $r$ , it continues to be the node with the minimum frequency in round  $r+1$*

*Proof:* The proof is provided in Section VIII-H. ■

Let  $\xi^*\{r\}$  denote the maximum difference in frequencies between any two nodes in round  $r$ :  $\xi^*\{r\} = F_{max}\{r\} - F_{min}\{r\}$ . By definition,  $\xi^*\{r\} \geq 0$ . We show that if there is any node that is “very far” away from convergence in round  $r$ ,  $\xi^*$  is guaranteed to decrease by  $\mu$ .

**Proposition 5.** *If  $S_1\{r\}$  or  $S_{-1}\{r\} \neq \emptyset, \xi^*\{r+1\} \leq \xi^*\{r\} - \mu$ . Furthermore, if both  $S_1\{r\}$  and  $S_{-1}\{r\} \neq \emptyset, \xi^*\{r+1\} = \xi^*\{r\} - 2\mu$ .*

*Proof:* Consider the case when  $S_1\{r\}$  and  $S_{-1}\{r\} \neq \emptyset$ . Then, from Proposition 4,  $F_{max}\{r+1\} = F_{max}\{r\} - \mu$  and  $F_{min}\{r+1\} = F_{min}\{r\} + \mu$ . Therefore,  $\xi^*\{r+1\} = F_{max}\{r+1\} - F_{min}\{r+1\} = (F_{max}\{r\} - \mu) - (F_{min}\{r\} + \mu) = \xi^*\{r\} - 2\mu$ .

Now consider the case when  $S_1 \neq \emptyset$  but  $S_{-1} = \emptyset$  (the reversed case is very similar). From Proposition (4),  $F_{max}\{r+1\} = F_{max}\{r\} - \mu$ . Similarly, from Proposition (3),  $F_{min}\{r+1\} \geq F_{min}\{r\}$ . Therefore, we can see that  $\xi^*$  must decrease at least by  $\mu$  as follows:  $\xi^*\{r+1\} = F_{max}\{r+1\} - F_{min}\{r+1\} = (F_{max}\{r\} - \mu) - F_{min}\{r+1\} \leq (F_{max}\{r\} - \mu) - F_{min}\{r\} = \xi^*\{r\} - \mu$ . ■

We now tackle the scenario when there is no node very far from convergence. To do this, we need a preliminary result which shows that the mean frequency does not change a lot across rounds.

**Corollary 1.** *The mean frequency  $\bar{\psi}\{r+1\}$  is bounded below by  $\bar{\psi}\{r\} - \mu$  and above by  $\bar{\psi}\{r\} + \mu$ .*

*Proof:* Since  $\mathcal{N}_i$  changes its frequency by at most  $\mu$ ,  $F_i\{r\} - \mu \leq F_i\{r+1\} \leq F_i\{r\} + \mu$ . Adding these inequalities across nodes and dividing by the number of nodes, we get  $\bar{\psi}\{r\} - \mu \leq F_i\{r+1\} \leq \bar{\psi}\{r\} + \mu$ . ■

Next, we show that if a node is “close to convergence” (within  $\epsilon + \mu + \chi$  of the mean frequency) in round  $r$ , *status quo* will prevail in round  $r+1$ : it will be no further than

$\epsilon + \mu + \chi$  away from the mean frequency in round  $r + 1$  too. This result guarantees that once the node frequencies have converged to within  $\epsilon + \mu + \chi$  of the mean, they will not drift away in subsequent rounds.

**Proposition 6.** *If  $\mathcal{N}_i \notin S_1\{r\} \cup S_{-1}\{r\}$  so that  $|\delta_i\{r\}| \leq (\epsilon + \mu + \chi)$ , then  $|\delta_i\{r+1\}| \leq (\epsilon + \mu + \chi)$ .*

*Proof:* The proof is provided in Section VIII-I ■

Let  $\delta^*\{r\} = \max_i |\delta_i\{r\}|$  denote the maximum excess frequency across nodes in round  $r$ . We put Propositions (5) and (6) together, show that  $\delta^*\{r\}$  is eventually no bigger than  $\epsilon + \mu + \chi$  and demonstrate the convergence of our algorithm with the averaged system.

**Theorem 1.** *In each round  $r$ , atleast one of the following statements is true,*

- 1)  $\xi^*\{r+1\} \leq \xi^*\{r\} - \mu$ .
- 2)  $\delta^*\{r\}$  and  $\delta^*\{r+1\} \leq (\epsilon + \mu + \chi)$

*Therefore, eventually, the maximum deviation in frequency from the mean is atmost  $\epsilon + \mu + \chi$ .*

*Proof:* In the  $r^{\text{th}}$  round of frequency adjustments,  $S_{far}\{r\} \triangleq S_1\{r\} \cup S_{-1}\{r\}$  is either empty or it is non-empty. If  $S_{far}\{r\}$  is nonempty, then by Proposition (5),  $\xi^*\{r+1\} \leq \xi^*\{r\} - \mu$ . On the other hand, if  $S_{far}\{r\}$  is empty, all the nodes are in  $(\cup_{i=2}^5 S_i\{r\}) \cup (\cup_{i=2}^5 S_{-i}\{r\})$ . Thus, the excess frequency of each node is smaller (in magnitude) than  $(\epsilon + \mu + \chi)$  and hence,  $\delta^*\{r\} \leq (\epsilon + \mu + \chi)$ . In this scenario, Proposition (6) guarantees that  $|\delta_i\{r+1\}| \leq (\epsilon + \mu + \chi) \forall i \Rightarrow \delta^*\{r+1\} \leq (\epsilon + \mu + \chi)$ . Using recursion, we can conclude that if  $\delta^*\{r\} \leq (\epsilon + \mu + \chi)$ , then  $\delta^*\{r'\} \leq (\epsilon + \mu + \chi) \forall r' \geq r$ .

We have already seen that  $\xi^*\{r\} \geq 0 \forall r$ . Assume that the node frequencies are bounded by  $1 \pm \rho_{max}$  before any frequency adjustments are made. Therefore,  $\xi^*\{1\} = F_{max}\{1\} - F_{min}\{1\} \leq 2\rho_{max}$ . If condition (1) is true in round  $r$ , we know that  $\xi^*$  decreases by  $\mu$ . However, condition 1 cannot be true indefinitely because  $\xi^*\{r\}$  is lower bounded by 0 and  $\xi^*\{1\}$  is finite. Therefore, condition 2 will have to be true in some round  $r = R_0$  for the first time. By our proof, it will then continue to hold forever. Therefore, we can conclude that  $\delta^*\{r'\} \leq \epsilon + \mu + \chi \forall r' \geq R_0$ . ■

We can now estimate  $R_0$  - the number of rounds needed for the maximum excess frequency to become smaller than  $\epsilon + \mu + \chi$  for the first time. By definition, we have,  $\xi^*\{R_0\} \leq 2\delta^*\{R_0\} / \leq 2(\epsilon + \mu + \chi)$ . In each round  $r$  between 1 and  $R_0$ , we know from Theorem (1) that  $\xi^*$  decreases by  $\mu$  (at least). Therefore,  $R_0$  needs to be no bigger than  $\frac{2\rho_{max} - 2(\epsilon + \mu + \chi)}{\mu}$ . Note that, while the number of *slots* needed per round of frequency adjustment might (and typically does) increase with network size, the number of *rounds* required for convergence is independent of the number of nodes in the network. We now use these insights to design an algorithm for the actual system that achieves frequency and phase synchrony.

## V. LLN ARGUMENTS

In order to map from the averaged system to the actual system, we realize that each step in the averaged system is effectively equivalent to many steps in the actual system, based on LLN-style arguments. Motivated by the algorithm for the averaged system, nodes in the actual system adjust their phases each time they receive a packet, but adjust their frequencies only occasionally - say, once in a round of  $S_R$  slots. However, the frequency adjustments are not based on phase errors in any particular slot; instead, they rely on averages of phase errors with neighbors over “many” slots. This lets us average over the randomness in the schedule by exploiting the LLN. We now explain how the nodes can estimate their excess frequencies at the end of the first round of slots, with the understanding that a similar procedure is repeated in future rounds.

Consider a slot  $s$  in the first round ( $0 \leq s \leq S_R - 1$ ). Since our frequency adjustments are based on averages of phase errors, we maintain a running average of the phases from the start of the round to any slot  $s < S_R$ :  $\varphi_{av}[s] = \sum_{s'=0}^s \varphi[s'] / (s + 1)$ . Before we obtain an expression for  $\varphi_{av}[s]$ , we make an approximation. Guided by equation (12), we approximate the product of the stochastic matrices  $G_s G_{s-1} \dots G_0$  by a rank-one matrix of the form  $\mathbf{1}\gamma^T$  when the number of matrices in the product exceeds a critical limit  $S_w$ . The critical limit  $S_w$  depends on the set  $\{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_M\}$  from which the matrices  $\{G_t\}$  are chosen and the probabilities  $\{p_1, p_2, \dots, p_M\}$  with which they are chosen from this set. Using this approximation and substituting the expression for the phases in slot  $s$  from equation (11), we get the averaged phases in the last slot of the first round - slot  $S_R - 1$  - to be,

$$\begin{aligned} \varphi_{av}[S_R - 1] = & \frac{[\sum_{s'=2}^{S_R-1} s' \bar{\psi}[0] + \sum_{s'=S_w}^s (\gamma^T \varphi[0])]}{S_R} \mathbf{1} \\ & + \frac{\varphi[0] + \varphi[1] + \sum_{s'=2}^{S_w-1} G_{s'-1} G_{s'-2} \dots G_0 \varphi[0]}{S_R} + \\ & \frac{\sum_{s'=2}^{S_R-1} (\mathbb{I} + \sum_{p=1}^{s'-1} G_{s'-1} G_{s'-2} \dots G_{s'-p})}{S_R} \delta[0] \quad (23) \end{aligned}$$

As before, we split the time averaged phases  $\varphi_{av}$  into a mean  $\bar{\varphi}_{av}$  and an excess part  $\varphi_{av,ex}$  so that  $\varphi_{av} = \bar{\varphi}_{av} \mathbf{1} + \varphi_{av,ex}$  and  $\mathbf{1}^T \varphi_{av,ex} = 0$ . The first term in equation (23) can be discarded since it only contributes to the mean phase  $\bar{\varphi}_{av}$  and our algorithm uses only the *excess-averaged-phases*  $\varphi_{av,ex}$ . The second term is a transient that vanishes when the number of slots  $S_R$  is large. Therefore, only the third term contributes to  $\varphi_{av,ex}$ . We simplify the third term and derive an expression for the excess-averaged-phases  $\varphi_{av,ex}$  in Section VIII-J. We now summarize this result. Let  $S_{\bar{\mathcal{G}}}$  be the smallest number so that  $\lambda_2^{S_{\bar{\mathcal{G}}}}$  is “negligible”, where  $\lambda_2$  is the second largest eigenvalue of the averaged system matrix  $\bar{\mathcal{G}}$ . Then, we can approximate the excess-averaged-phases  $\varphi_{av,ex}$  at the end of the first round by,

$$\varphi_{av,ex}[S_R - 1] \approx (\mathbb{I} - \bar{\mathcal{G}}_{ex})^{-1} \delta[0]. \quad (24)$$

by choosing  $S_R$  to be much larger than the critical limit  $S_w$  and  $S_{\bar{\mathcal{G}}}$ . Here,  $\bar{\mathcal{G}}_{ex}$  is the excised version of the system matrix for the averaged system. Equivalently, the skews  $\delta[0]$  can be estimated as  $\delta[0] \approx (\mathbb{I} - \bar{\mathcal{G}}_{ex})\varphi_{av,ex}$ . Of course, we do not have access to the excess-averaged-phases  $\varphi_{av,ex}$ . But, we can use the fact that  $\bar{\mathcal{G}}$  is a stochastic matrix and mimic the proof for the averaged system (see Section VIII-D) to show that  $(\mathbb{I} - \bar{\mathcal{G}}_{ex})\varphi_{av,ex} = (\mathbb{I} - \bar{\mathcal{G}})\varphi_{av}$ . Thus, the nodes can estimate their excess frequencies from the raw phases as  $\delta[0] \approx (\mathbb{I} - \bar{\mathcal{G}})\varphi_{av}[S_R - 1]$ . Let us denote the averaged phase at node  $i$  in slot  $S_R - 1$  by  $\varphi_{av,i}[S_R - 1]$ . By substituting the expression for  $\bar{\mathcal{G}}$  (and mimicking the proof in Section VIII-E), we get the skew estimate of the  $i$ th node to be,

$$\hat{\delta}_i = \beta \sum_{j \neq i} q_{j \rightarrow i} (\varphi_{av,i}[S_R - 1] - \varphi_{av,j}[S_R - 1]) \quad (25)$$

where  $q_{j \rightarrow i}$  is the probability that node  $j$  transmits to node  $i$ . We now explain how node  $i$  can implement this rule in a simple, distributed fashion without any communication overhead. From the definition of the averaged phases, we have,

$$\varphi_{av,i}[S_R - 1] - \varphi_{av,j}[S_R - 1] = \sum_{s=0}^{S_R-1} \frac{\varphi_i[s] - \varphi_j[s]}{S_R} \quad (26)$$

Therefore,  $\varphi_{av,i}[S_R - 1] - \varphi_{av,j}[S_R - 1]$  depends on the phase error between  $\mathcal{N}_i$  and  $\mathcal{N}_j$  in all slots between 0 and  $S_R - 1$ . However,  $\mathcal{N}_i$  cannot measure all these phase errors; it can only estimate the errors in those slots when it receives a packet from  $\mathcal{N}_j$ . But,  $\mathcal{N}_i$  can approximate  $\varphi_{av,i}[S_R - 1] - \varphi_{av,j}[S_R - 1]$  by averaging over the phase errors that it does measure. Assume that  $\mathcal{N}_j$  that has a nonzero probability of transmitting a packet to  $\mathcal{N}_i$  (i.e.  $q_{j \rightarrow i} > 0$ ) and that it transmits on  $n_{ij}$  occasions in the first round of slots to  $\mathcal{N}_i$ . We denote these slots by  $s_{j \rightarrow i}(1), s_{j \rightarrow i}(2), \dots, s_{j \rightarrow i}(n_{ij})$  respectively. Therefore,  $\mathcal{N}_i$  approximates the average phase error over the first round with  $\mathcal{N}_j$  to be,

$$\varphi_{av,i}[S_R - 1] - \varphi_{av,j}[S_R - 1] \approx \sum_{t=1}^{n_{ij}} \frac{\varphi_i[s_{j \rightarrow i}(t)] - \varphi_j[s_{j \rightarrow i}(t)]}{n_{ij}} \quad (27)$$

We use the empirical definition of probability to approximate  $q_{j \rightarrow i}$  as  $q_{j \rightarrow i} \approx n_{ij}/S_R$  and substitute equation (27) in equation (25) to get,

$$\hat{\delta}_i[S_R - 1] \approx \frac{\beta}{S_R} \sum_{j \neq i} \sum_{t=1}^{n_{ij}} \left( \varphi_i[s_{j \rightarrow i}(t)] - \varphi_j[s_{j \rightarrow i}(t)] \right) \quad (28)$$

We use this as the defining equation for  $\mathcal{N}_i$ 's estimate of its excess frequency: it simply adds the phase errors it observes when it receives packets from its neighbors over the entire round (and scales it by  $\beta/S_R$ ) to decide on the frequency adjustment at the end of the round. This rule is intuitively pleasing: if a node finds that despite all the phase adjustments made, its clock is still "ahead" ( $\varphi_i - \varphi_j > 0$ ) of its neighbors' clocks over a long time, it concludes that its clock is running faster than the average across the network. We now summarize the algorithm from the point of view of node  $i$ .

**Overall algorithm:** Node  $i$  estimates its skew over a round according to equation (28). If  $\hat{\delta}_i$  falls within a dead zone of width  $\epsilon_{approx}$  (i.e.  $|\hat{\delta}_i| \leq \epsilon_{approx}$ ), node  $i$  does not change its frequency; in the other cases, node  $i$  changes its frequency by  $\pm\mu$  based on the sign of  $\hat{\delta}_i$  i.e. it makes a frequency jump  $\Delta F_i = -\mu \text{sign}(\hat{\delta}_i)$  at the end of a round of  $S_R$  slots. In subsequent rounds, the process is repeated by resetting the time-averaged phases and recomputing them from the start of the round.

An attractive feature of the frequency adjustment rule is the minimal storage required at each node, in spite of the significant amount of memory the rule entails. To adjust its frequency, each node needs to store only one number and recursively update it. Thus, the storage needed is independent of the size of the network and the network topology.

**Tackling Measurement Noise:** When the implicit timestamps are noisy, we leave the phase adjustment rule unaltered but change the frequency adjustment rule. We now describe these modifications. Let  $\xi_{ij}[s_{j \rightarrow i}(t)] = \varphi_i[s_{j \rightarrow i}(t)] - \varphi_j[s_{j \rightarrow i}(t)]$  denote the true phase offset between  $\mathcal{N}_i$  and  $\mathcal{N}_j$  in slot  $s_{j \rightarrow i}(t)$  and  $\hat{\xi}_{ij}[s_{j \rightarrow i}(t)]$  denote  $\mathcal{N}_i$ 's estimate of this offset. The phase offset estimation error  $\hat{\xi}_{ij}[s_{j \rightarrow i}(t)] - \xi_{ij}[s_{j \rightarrow i}(t)]$ , induced by the measurement noise is denoted by  $\gamma_{ij}[s_{j \rightarrow i}(t)]$ . From equation (28), this translates to an error of  $\Delta e_i = \frac{\beta}{S_R} \sum_{j=1}^N \sum_{t=1}^{n_{ij}} \gamma_{ij}[s_{j \rightarrow i}(t)]$  in  $\mathcal{N}_i$ 's estimate of its excess frequency. We model the errors in estimating the phase offset  $\gamma_{ij}[s_{j \rightarrow i}(t)]$ , as random variables that are independent across slots and node-pairs. Furthermore, we assume that these errors are picked from the same distribution  $f_{\Gamma}(\gamma)$  with zero mean and a standard deviation  $\sigma_f$ . For example, in our simulations, we choose  $f_{\Gamma}(\gamma)$  to be uniform. Let us assume that  $\mathcal{N}_i$  receives  $P_i = \sum_j n_{ij}$  packets, from all its neighbors combined, in a round of  $S_R$  slots. We can rewrite the error in estimating the excess frequencies due to the measurement noise as  $\Delta e_i = \frac{\beta P_i}{S_R} \times \frac{1}{P_i} \sum_{j=1}^N \sum_{t=1}^{n_{ij}} \gamma_{ij}[s_{j \rightarrow i}(t)]$ . If  $P_i$  is reasonably large, we can invoke the Central Limit Theorem to approximate the distribution of  $\frac{1}{P_i} \sum_j \sum_t \gamma_{ij}[s_{j \rightarrow i}(t)]$  by a Gaussian with zero mean and variance  $\sigma^2 = \sigma_f^2/P_i$ . Since a Gaussian random variable is very likely to be contained within three standard deviations on either side of the mean, we approximate the estimation error induced by the noise  $\Delta e_i$  to be bounded in magnitude by  $(\beta P_i/S_R) \times (3\sigma) = (\beta P_i/S_R) \times (3\sigma_f/\sqrt{P_i})$ . To combat this error in the estimate of its excess frequency,  $\mathcal{N}_i$  further increases the size of the dead zone from its value in the noiseless scenario. This increase needs to be no larger than the maximum value that the error can take, or  $\frac{\beta P_i}{S_R} 3\sigma_f/\sqrt{P_i}$ . However, in our simulations we are more conservative and increase the width of dead zone by  $3\sigma_f/\sqrt{P_i}$  (note that  $\beta P_i/S_R$  is guaranteed to be smaller than 1). To summarize, the only change in the frequency adjustment rules is an increase in the width of the dead zone;  $\mathcal{N}_i$ 's dead zone in the presence of measurement noise with variance  $\sigma_f^2$  given by  $\epsilon'_i = \epsilon_{approx} + 3\sigma_f/\sqrt{P_i}$  where  $\epsilon_{approx}$  is the dead-zone width in the noiseless setting and  $P_i$  is the number of packets received by  $\mathcal{N}_i$  over a round of slots.

## VI. SIMULATION RESULTS

We now describe the models and parameters used in our simulations, followed by the results.

**Topologies:** We consider the *ring* topology (each node has two neighbors) and the rectangular *grid* topology (each node has four neighbors) as canonical examples of two-dimensional networks with large and small diameter, respectively.

**Interference Model:** We use the node exclusive interference model (two links can be active if they do not have a node in common) as an abstraction for “highly directional” networks (such as mm-wave networks [1]). For omnidirectional networks, we use the 2-hop interference model [23]: two links can be active simultaneously as long as they do not have a node in common, and the nodes involved in the two links are not neighbors.

**Communication Patterns:** We consider TDM schedules in which a randomly chosen maximal matching is active in each slot; a *matching* is a set of links that can be simultaneously active without interfering with one another, and a *maximal matching* is a matching to which no link can be added without interfering with one of the active links. For large networks (e.g., a 64 node grid with directional links), it is infeasible to enumerate all maximal matchings. In this case, for each link, we randomly choose 120 maximal matchings in which it participates. The overall set of maximal matchings  $\mathcal{S}_{\text{matching}}$  is obtained by taking the union of the sets generated for each link.

**Opportunistic listening:** In omnidirectional networks, we consider two modes for timing adjustment. In the ONLYINTENDED mode, a node adjusts its clock only when it receives a packet explicitly addressed to it. In the EAVESDROP mode, a node adjusts its clock whenever it can receive a collision-free packet, even if it is not the intended recipient.

**Simulation Parameters:** We fix  $\beta = 0.5$  in all our simulations. Our choice of timescales in directional networks are motivated by Gigabit rate mm-wave networks: we choose the slot time  $T_{\text{slot}} = 10\mu\text{s}$  and for the coarse synchronization procedure at startup, we choose  $\tau_{\text{min}} = 10\mu\text{s}$  and  $\tau_{\text{max}} = 300\mu\text{s}$  (see Section II). The timescales in omnidirectional settings are motivated by WiFi-style networks: we choose  $T_{\text{slot}} = 10\text{ms}$  and  $\tau_{\text{min}} = 10\text{ms}$  and  $\tau_{\text{max}} = 300\text{ms}$ . We set the maximum value of the skew  $\rho_{\text{max}}$  to 50 ppm.

**Performance Metric:** Since TDM slotting overhead depends on the phase error among communicating nodes, the key performance metric is the *worst neighbor timing error*, which is the largest magnitude of phase error *between neighbors*.

### A. Phase-Only Adjustments

We consider directional networks and omnidirectional networks in the ONLYINTENDED mode consisting of 9-64 nodes set in ring and grid topologies. For each topology, we generate a set of “bad” skews by solving linear programs that optimize the phase error on each link of the averaged system. We choose these “bad” skews to be the distribution of skews across nodes in the actual system. We then average the worst error between nodes in the actual system at the end of 3000 slots over 2000

realizations of communication patterns. The results for the directional and omnidirectional settings are shown in Figures 3 and 4 respectively. The figures confirm that the errors for the averaged system are indeed a lower bound for those in the actual system, and that the worst neighbor error increases with network size. The error grows faster with  $N$  for a ring topology. For directional ring networks (Figure 3(a)), the error grows as  $N$  in the averaged system and as  $N^{1.33}$  (approximately) in the actual system. For directional grid networks (Figure 3(b)), the error grows only as  $N^{0.66}$  in the averaged system and as  $N^{0.875}$  (approximately) in the actual system.

**Typical numbers for directional networks:** From Figure 3, we see that the error between neighbors in a 64 node network set in a ring topology can be as large as 220 ns. This is comparable in magnitude to the guard interval needed to handle propagation delays. For example, the envisioned Gigabit rate outdoor mesh networks with link ranges on the order of 200 m require guard intervals of  $200\text{m}/(3 \times 10^8\text{m/s}) \approx 666\text{ns}$  to handle propagation delays. Therefore, an additional synchronization error of 220 ns represents a 33% increase in the overhead. On the other hand, for small directional networks in a ring topology (say, 16 nodes) or reasonably large networks in a grid topology (36 nodes), we see that the largest phase error between neighbors will only be 40 ns. Therefore, the maximum increase in the overhead is only 6 %. To summarize, phase-only adjustments with implicit timestamps may suffice in small directional networks with linear topologies or moderately sized networks in grid topologies, but frequency adjustments are necessary for large networks, especially in linear topologies.

**Recommendations for omnidirectional networks:** Since the slot duration  $T_{\text{slot}}$  is longer in omnidirectional networks and the phase error scales in proportion to  $T_{\text{slot}}$ , we see from Figure 4 that the errors with phase-only adjustments are also correspondingly larger ( $\sim 10$ 's of  $\mu\text{s}$  even for small networks). However, the guard interval needed to handle propagation delays is still  $\approx 1\mu\text{s}$  since the link ranges are on the order of hundreds of meters. Therefore, phase-only adjustments are not sufficient to guarantee phase synchrony on the order of the guard interval in omnidirectional networks. However, a more practical definition of the tolerable overhead is to define it as a fraction of the payload (slot size) rather than the guard interval for propagation delay. Phase-only adjustments might then suffice since the phase errors grow in proportion to  $T_{\text{slot}}$  and their *ratio* is simply a constant. Thus, here too, phase-only adjustments might suffice for small-to-moderate sized networks.

### B. Phase & Frequency Adjustments

We begin by introducing parameters and performance metrics specific to the phase-frequency adjustment algorithm and then describe the results.

**Simulation Parameters:** We consider two types of skew distribution across nodes: (a) “random skews” chosen independently and uniformly in  $[-50\text{ppm}, 50\text{ppm}]$ ; (b) “bad skews” chosen by maximizing the worst neighbor error for the aver-

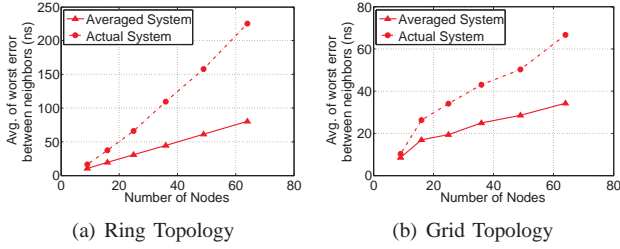


Fig. 3. Worst error between neighbors for the actual system and the averaged system with only phase adjustments in a *directional network*.

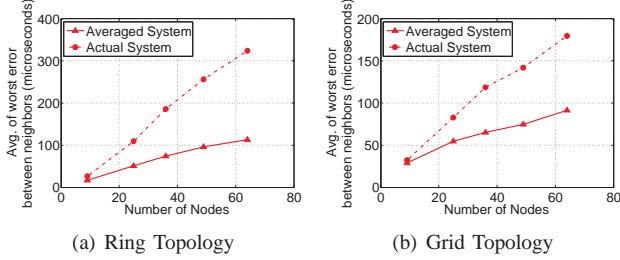


Fig. 4. Worst error between neighbors for the actual system and the averaged system with only phase adjustments in an *omnidirectional network*. The network operates in the ONLYINTENDED mode.

aged system. Nodes adjust their frequencies by  $\mu = \pm 1 \text{ppm}$  once a round, which consists of  $S_R = 200$  slots. We investigate the performance with noiseless as well as noisy implicit timestamps. We model the noise as being uniformly distributed in  $[-5ns, 5ns]$  for directional networks and in  $[-5\mu s, 5\mu s]$  for omnidirectional networks. We use 30,000 TDM slots for each simulation run (and average over 50 runs). We choose the width of the dead zone in the noiseless setting  $\epsilon_{approx}$  to be  $3\mu = 3 \text{ppm}$ .

**Performance Metrics:** In addition to the worst phase error between neighbors, we also investigate the *network wide frequency error* as a measure of frequency synchrony. The network wide frequency error is defined to be the maximum difference in frequency between any two nodes in the network. We now describe the typical evolution of node frequencies in a single trial and then provide the results for the phase and frequency errors averaged over 50 trials.

**Evolution of node frequencies:** We describe three scenarios that serve as exemplars for the evolution of frequencies at different nodes in a single simulation run. *Example 1:* We consider a 36 node omnidirectional network arranged in a grid topology. The skews at different nodes are distributed randomly and the implicit timestamps are noiseless. From Figure (5), we see that nodes begin with varying frequencies and move in concert towards a common frequency. Every node's frequency is eventually confined to the band  $f_{nom} \times [1 - 1.25 \text{ppm}, 1 + 0.34 \text{ppm}]$  where  $f_{nom}$  is the nominal frequency. Note that  $f_{nom}$  itself is not within this band. This shows that the nodes simply converge to a common frequency and not to the nominal frequency. Even in the noiseless scenario, the eventual frequency alignment is not

perfect for two reasons: the adjustments have a resolution of  $\pm 1 \text{ppm}$  and we have enforced a dead zone within which nodes do not change their frequencies. *Example 2:* A bad skew distribution best illustrates the fact that the nodes adjust their frequencies in concert. We consider such a skew distribution in a 36 node directional network with a ring topology. The implicit timestamps are noiseless in this example. We plot the frequencies of *all* nodes as a function of time in Figure (6). However, we see only two traces until about 10000 slots. This is because the nodes start off at identical frequencies of  $\pm 50 \text{ppm}$  and make identical frequency adjustments of  $\mp 1 \text{ppm}$  in each round. Consequently, at any time, the frequencies of all the nodes are confined to one of two values,  $f_s(1 \pm k)$ ,  $k \in \{1, 2, \dots, 50\} \text{ppm}$ . We also see that some nodes make unnecessary adjustments when the frequencies have almost converged (around 10000 slots). If the dead zone is not large enough, such unnecessary adjustments by a few nodes could drag the frequencies of the other nodes along. This might result in the nodes eventually having a common frequency that is outside the band where they began i.e  $f_s \times [1 - \rho_{max}, 1 + \rho_{max}]$ . This is not desirable in practice. In our simulations, we find that a dead zone of width  $3\mu$  suffices in the noiseless setting to prevent this from happening. A precise characterization dead zone width required to guarantee that the frequencies remain bounded within the original range, similar to the one provided for the averaged system, is left as an open issue. *Example 3:* We now consider the same

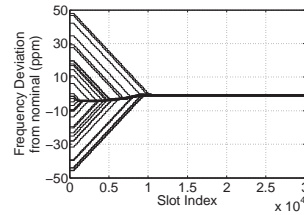


Fig. 5. Frequency deviations of all 36 nodes in an *omnidirectional network* with a *grid topology*. Skews are *randomly distributed* and measurements are *noiseless*.

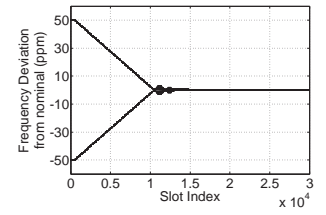


Fig. 6. Frequency deviations of all 36 nodes in a *directional network* with a *ring topology*. Skews are *distributed adversely* and measurements are *noiseless*.

setting as in Example 1, except that the measurements are noisy. From Figure 7, we see that the steady march of the nodes towards convergence is eventually stalled by the measurement noise. The node frequencies fall within the dead zone and the nodes are unable to make a reliable decision on whether their frequencies are greater or lesser than the average. Thus, we have an eventual network-wide frequency error of 11.7 ppm in this case. The size of the dead zone could potentially be reduced by waiting longer between frequency adjustments, thereby allowing the nodes to average the noise in the timestamps to a greater degree. However, this approach does not provide substantial gains because the phase error between nodes, in this regime, is dominated by erroneous phase adjustments rather than mildly disparate frequencies.

While we have limited our discussion here to three repre-

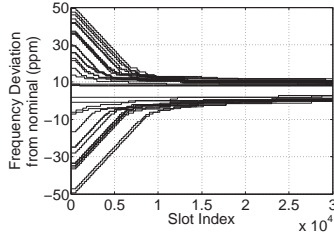


Fig. 7. Frequency deviations of all 36 nodes in an omnidirectional network with a grid topology. Skews are randomly distributed and measurements are noisy.

sentative examples due to space restrictions, we have carried out extensive simulations that show that our observations apply qualitatively to diverse combinations of node topologies, network directionality and measurement noise strength.

**Mean Network Wide Frequency Error:** We plot the network wide frequency error, averaged over 50 runs, as a function of time in Figures 8 - 11. We consider networks of 16/36/64 nodes arranged in a ring/grid topology with noiseless and noisy implicit timestamps. We examine each scenario for omnidirectional as well as directional networks. Two things are common to these plots: (1) For about the first 10000 slots, the network wide frequency error drops by  $2\mu = 2$  ppm each time a frequency adjustment is made. This drop is virtually independent of different parameters such as network size, topology, directionality and measurement noise. The steady fall occurs because there are nodes with frequencies far above and below the mean at the beginning; these nodes make the right adjustment every time (see Proposition 5). (2) We observe that the number of rounds of frequency adjustment that contribute to a decrease in the network wide frequency error is also virtually independent of parameters such as network size, topology, directionality and measurement noise. It depends only on the network wide frequency error at the start and the adjustment step size. Since the network wide error drops by 2 ppm in each round, the number of rounds of frequency adjustment required is roughly  $100 \text{ ppm} / 2 \text{ ppm} = 50$ .

An important metric is the eventual network wide frequency error. For the noiseless setting (Figures 8(a), 9(a), 10(a), 11(a)) the network wide frequency error settles to a value on the order of the adjustment step-size  $\mu = 1$  ppm. This value is almost independent of the network topology and the directionality; for all combinations of the topology and directionality, the error is confined to 0.9-1.2 ppm. When the implicit timestamps are noisy, the eventual network wide frequency error depends mildly on the size of the network and the directionality (Figures 8(b), 9(b), 10(b), 11(b)). For example, in directional networks with a ring topology, the eventual network wide frequency error is about 8.9 ppm with 16 nodes and it increases to 10.2 ppm with 64 nodes; the corresponding numbers for an omnidirectional setting are 9.4 ppm and 10.9 ppm respectively. The eventual frequency synchronization error is also a little higher with the grid topology: for example, in a directional setting, the error is about 10.1 ppm with 16 nodes and 13.3

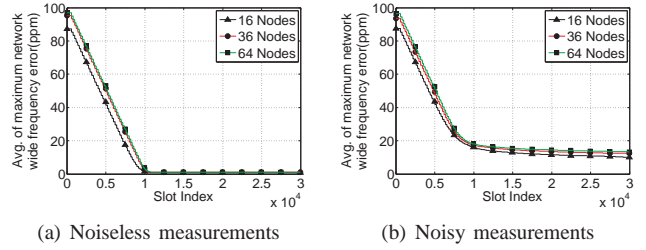


Fig. 8. Network wide frequency error in a directional setting with a grid topology. Skews are distributed randomly.

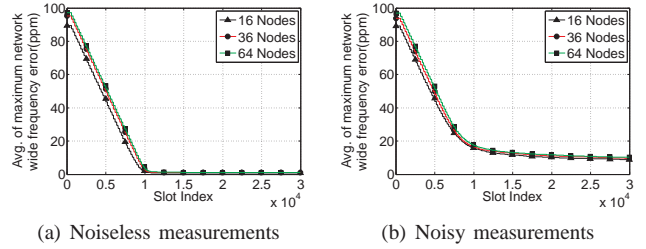


Fig. 9. Network wide frequency error in a directional setting with a ring topology. Skews are randomly distributed.

ppm with 64 nodes. In all these cases, we see that there is a ten-fold decrease in the network wide frequency error - from 100 ppm to about 10 ppm (roughly).

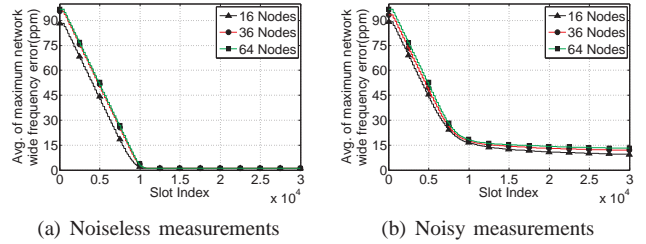


Fig. 10. Network wide frequency error in an omnidirectional setting with a grid topology. Skews are randomly distributed.

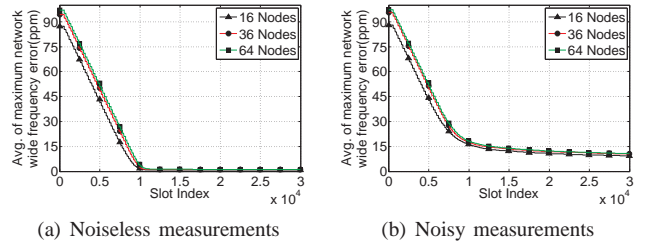


Fig. 11. Network wide frequency error in an omnidirectional setting with a ring topology. Skews are randomly distributed.

**Worst Phase Error Between Neighbors:** We now investigate the largest error in clock phase between any two nodes that are neighbors. We split our discussion into two cases: in the first case, the skews are distributed randomly and in the second case, they are distributed badly.

*Case A - Random Distribution of Skews:* We choose the

node skews randomly and plot the worst phase error for directional networks in Figure 12. We also plot the worst phase error for omnidirectional networks operating in the ONLYINTENDED mode and EAVESDROP mode in Figures 13 and 14 respectively. Note that all the plots have noisy implicit timestamp measurements. From these plots, we see that the phase errors fall “smoothly” from their values at startup - after the coarse synchronization procedure - to their steady-state values. The eventual phase error increases mildly with network size and changes very little with the topology. For example, in directional networks with a ring topology, the eventual phase error only increases from 6.25 ns to 10.7 ns when the network size increases from 16 nodes to 64 nodes. The corresponding numbers for the grid topology are very similar: 6.49 ns (16 nodes) and 9.05 ns (64 nodes). Similarly, the errors in omnidirectional networks with ring and grid topologies are close; they increase from  $6.5\mu s$  for 16 node networks to about  $10\mu s$  for 64 node networks.

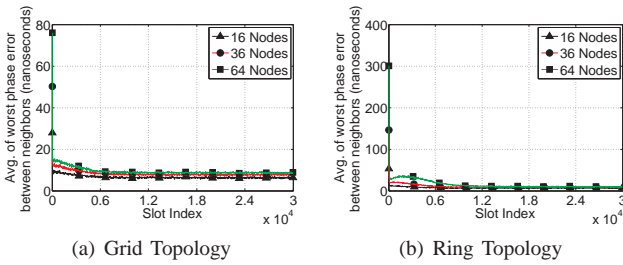


Fig. 12. Worst phase error between neighbors in a *directional* setting with *noisy* measurements. Skews are *randomly* distributed.

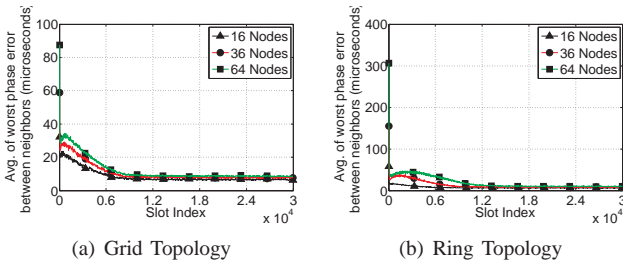


Fig. 13. Worst error between neighbors in an *omnidirectional* setting with *randomly* distributed skews. Measurements are *noisy* and nodes are in the ONLYINTENDED mode.

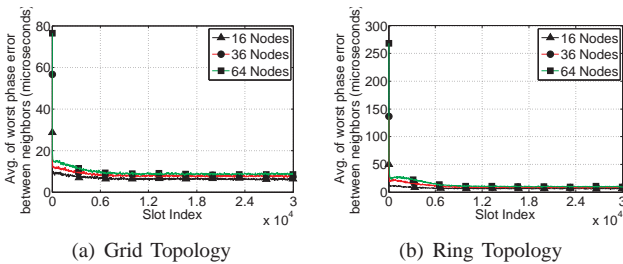


Fig. 14. Worst error between neighbors in an *omnidirectional* setting with *randomly* distributed skews. Measurements are *noisy* and nodes are in the EAVESDROP mode.

*Case B - Badly distributed skews:* We plot the largest phase error with badly distributed skews for directional networks in Figure 15 and omnidirectional networks in Figures 16 and 17 respectively. These curves have a conspicuous “hump” at the start - particularly the ones corresponding to the ring topology, Figures 15(b), 17(a), 17(b)) - where the phase error rises before falling. This occurs because the round sizes (and hence spacing between frequency updates) required are large, which gives phase errors a chance to grow before they can be reduced with frequency updates. We now compare the size of the hump - which is the peak value of the phase error between neighbors - for differing network sizes, topologies and modes of operation.

Firstly, by comparing the plots for grid and ring topologies (for example, Figure 15(a) vs. 15(b) [or] Figure 16(a) vs. 17(a)), we observe that the hump is much larger with the ring topology. For example, in the ONLYINTENDED mode in omnidirectional networks with 64 nodes, the size of the hump is  $360\mu s$  in the ring topology as compared to  $175\mu s$  in the grid topology. This is consistent with earlier results from phase-only adjustments, where we found that the errors scale much faster with the number of nodes for a ring topology as compared to the grid topology. In all these figures, we see that the size of the hump increases with the number of nodes in the network. We can summarize both these facts by concluding that the size of the hump increases with the network diameter. Next, we compare omnidirectional networks operating in the EAVESDROP and ONLYINTENDED modes. Comparing Figures 17(a) and 17(b), we see that the size of the hump in an omnidirectional setting with a ring topology is almost twice as large in the ONLYINTENDED mode as compared to the EAVESDROP mode. For example, with a 64 node ring network, the size of the hump is about  $360\mu s$  in the ONLYINTENDED mode; but, it is only about  $200\mu s$  in the EAVESDROP mode. This is a clear illustration of the benefits of eavesdropping.

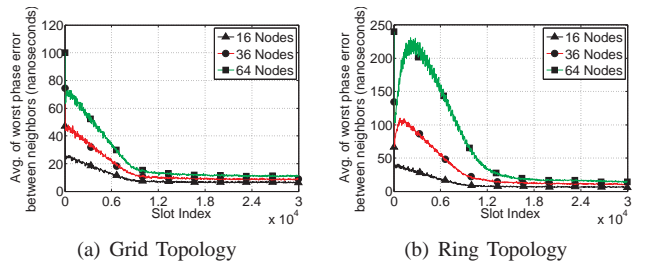


Fig. 15. Worst phase error between neighbors in a *directional* setting. Skews are *badly* distributed and measurements are *noisy*.

## VII. CONCLUSIONS

We have shown that it is possible to maintain timing synchronization without explicit signaling in a TDM-based mesh network. Our introduction of an averaged system enables us to provide a rigorous analysis and to give conditions for convergence, and our simulations show that the design prescriptions that we obtain for the averaged system do result in the desired performance for the original system. We have shown that our algorithm achieves phase synchrony on the order of tens

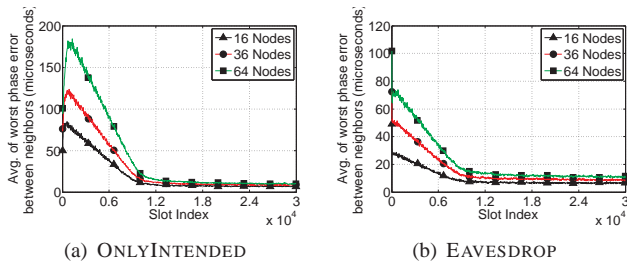


Fig. 16. Worst phase error between neighbors in an omnidirectional setting with a grid topology. Skews are badly distributed, measurements are noisy.

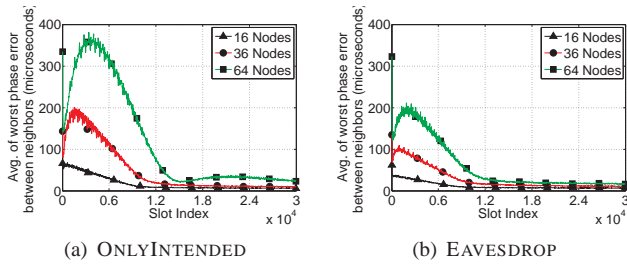


Fig. 17. Worst phase error between neighbors in an omnidirectional setting with a ring topology. Skews are badly distributed, measurements are noisy.

of microseconds for omnidirectional WiFi networks and tens of nanoseconds for directional 60 GHz networks. While the present paper establishes the basic feasibility of implicit timing synchronization, a significant amount of effort is required in translating our ideas to practice, starting with more detailed simulations that explicitly model traffic patterns, medium access control, and initial establishment of coarse timing synchronization (including estimation of propagation delays). In terms of implementation, a timing synchronization application can be layered on top of an existing network protocol stack as follows. The timing synchronization application would generate packets explicitly for synchronization only at network start-up, but would not generate packets for synchronization maintenance. Ideally, in order to obtain the most accurate measurements, the physical layer should communicate the implicit timestamps from received packets to this application, and the timing synchronization application should communicate its current estimates of phase and frequency to the physical layer so that the latter can adjust transmission times. However, it may be possible to obtain adequate performance even if the timing synchronization application communicates with the MAC layer, especially for outdoor settings in which propagation delays dictate larger TDM slot guard times, so that the requirements for synchronization accuracy are more relaxed. Fortunately, there are now a number of open source platforms on which such concepts can be experimentally validated, such as the WARP platform for physical layer synchronization, and the MadWiFi driver for MAC layer synchronization.

## REFERENCES

[1] S. Singh, R. Mudumbai, and U. Madhow, "Distributed coordination with deaf neighbors: efficient medium access for 60 GHz mesh networks," *IEEE INFOCOM 2010*, 2010.

[2] R. Mirollo and S. Strogatz, "Synchronization of pulse-coupled biological oscillators," *SIAM J. on Appl. Math.*, pp. 1645–1662, 1990.

[3] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, and R. Nagpal, "Firefly-inspired sensor network synchronicity with realistic radio effects," in *Proc. ACM SenSys 2005*, 2005, pp. 142–153.

[4] R. Solis, V. Borkar, and P. Kumar, "A new distributed time synchronization protocol for multihop wireless networks," in *Proc. of the 45th IEEE CDC*, 2006.

[5] L. Schenato and G. Gamba, "A distributed consensus protocol for clock synchronization in wireless sensor network," in *Proc. IEEE CDC 2007*.

[6] P. Sommer and R. Wattenhofer, "Gradient Clock Synchronization in Wireless Sensor Networks," in *Proc. ACM/IEEE IPSN*, 2009.

[7] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[8] W. Ren and E. Atkins, "Distributed multi-vehicle coordinated control via local information exchange," *International J. of Robust and Nonlinear Control*, vol. 17, no. 10–11, pp. 1002–1033, 2007.

[9] W. Ren, "Second-order consensus algorithm with extensions to switching topologies and reference models," in *American Control Conference, 2007. ACC'07*, 2007, pp. 1431–1436.

[10] A. Giridhar and P. Kumar, "Distributed time synchronization in wireless networks: Algorithms and analysis (I)," in *Proc. IEEE CDC 2006*.

[11] X. Gang and K. Shalinee, "Discrete-Time Second-Order Distributed Consensus Time Synchronization Algorithm for Wireless Sensor Networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, 2008.

[12] P. Baroah, J. Hespanha, and A. Swami, "On the effect of asymmetric communication on distributed time synchronization," in *Proc. 46th IEEE CDC*, 2007, pp. 5465–5471.

[13] O. Simeone, U. Spagnolini, Y. BarNess, and S. Strogatz, "Distributed synchronization in wireless networks," *IEEE Signal Processing Magazine*, vol. 25, no. 5, pp. 81–97, 2008.

[14] N. Freris and P. Kumar, "Fundamental limits on synchronization of affine clocks in networks," in *Invited paper in the 46th IEEE CDC*, 2007.

[15] L. Huang and T. Lai, "On the scalability of IEEE 802.11 ad hoc networks," in *Proc. 3rd ACM MOBIHOC*, 2002.

[16] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *ACM SIGOPS Operating Systems Review*, vol. 36, pp. 147–163, 2002.

[17] S. Ganeriwal, R. Kumar, and M. Srivastava, "Timing-sync protocol for sensor networks," in *Proceedings ACM SenSys 2003*. ACM New York, NY, USA, 2003, pp. 138–149.

[18] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi, "The flooding time synchronization protocol," in *Proc. ACM SenSys 2004*, 2004, pp. 39–49.

[19] W. Ren and R. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Transactions on Automatic Control*, vol. 50, no. 5, p. 655, 2005.

[20] S. Venkateswaran and U. Madhow. Distributed Implicit Timing Synchronization for Multihop Mesh Networks. [Online]. Available: <http://www.ece.ucsb.edu/wcs/Publications/implicitTimingSyncTR.pdf>

[21] G. Dantzig and M. Thapa, *Linear Programming: Theory and extensions*. Springer Verlag, 2003.

[22] R. Mudumbai, S. Singh, and U. Madhow, "Medium access control for 60 GHz outdoor mesh networks with highly directional links," *IEEE INFOCOM 2009*, 2009.

[23] G. Sharma, R. Mazumdar, and N. Shroff, "On the complexity of scheduling in wireless networks," in *Proc. ACM MobiCom 2006*.

[24] C. Meyer, *Matrix Analysis and Applied Linear Algebra*. SIAM, 2000.

## VIII. APPENDIX

In this section, we provide proofs of a number of statements in the paper.

### A. An expression for the excess phases $\varphi_{ex}[s]$ in the averaged system

We begin by obtaining an approximate expression for the phases in slot  $s$  (where  $s$  is large) by substituting equation (8)

into equation (6),

$$\varphi[s] \approx (\bar{\varphi}[0] + s\bar{\psi})\mathbf{1} + (\mathbb{I} + \sum_{k=1}^{s-1} \bar{\mathcal{G}}^k)\delta \quad (29)$$

We now simplify this expression by showing that  $\bar{\mathcal{G}}^k \delta = \bar{\mathcal{G}}_{ex}^k \delta$ . Since  $\mathbf{1}^T \delta = 0$ , we have

$$\bar{\mathcal{G}}^k \delta = \bar{\mathcal{G}}^k \delta - \frac{\mathbf{1}\mathbf{1}^T}{N} \delta = \left( \bar{\mathcal{G}}^k - \frac{\mathbf{1}\mathbf{1}^T}{N} \right) \delta \quad (30)$$

Using the spectral decomposition of  $\bar{\mathcal{G}}_{ex}$ , we see that,  $\bar{\mathcal{G}}_{ex}^k = \sum_{l=2}^N \lambda_l^k \mathbf{v}_l \mathbf{v}_l^T = \bar{\mathcal{G}}^k - \mathbf{1}\mathbf{1}^T/N$ . Substituting this into equation (30), we get  $\bar{\mathcal{G}}^k \delta = \bar{\mathcal{G}}_{ex}^k \delta$ . We now substitute this result back into equation (29) to obtain,

$$\varphi[s] \approx (\bar{\varphi}[0] + s\bar{\psi})\mathbf{1} + \sum_{k=0}^{s-1} \bar{\mathcal{G}}_{ex}^k \delta \quad (31)$$

Next, we compute the mean phase across nodes in slot  $s$ . We have,

$$\bar{\varphi}[s] = \frac{\mathbf{1}^T \varphi[s]}{N} = (\bar{\varphi}[0] + s\bar{\psi}) \frac{\mathbf{1}^T \mathbf{1}}{N} + \sum_{k=0}^{s-1} \mathbf{1}^T \bar{\mathcal{G}}_{ex}^k \delta \quad (32)$$

The term  $\mathbf{1}^T \mathbf{1}/N$  is simply 1. We now show that  $\mathbf{1}^T \bar{\mathcal{G}}_{ex}^k = \mathbf{0}^T$ . Since  $\bar{\mathcal{G}}_{ex}^k = \bar{\mathcal{G}}^k - \mathbf{1}\mathbf{1}^T/N$ , we get

$$\mathbf{1}^T \bar{\mathcal{G}}_{ex}^k = \mathbf{1}^T \bar{\mathcal{G}}^k - \frac{\mathbf{1}^T \mathbf{1}}{N} \mathbf{1}^T = \mathbf{1}^T \bar{\mathcal{G}}^k - \mathbf{1}^T \quad (33)$$

Since  $\bar{\mathcal{G}}$  is symmetric and stochastic, all its powers are also symmetric and stochastic. Thus, we have  $\mathbf{1}^T \bar{\mathcal{G}}^k = ((\bar{\mathcal{G}}^k)^T \mathbf{1})^T = (\bar{\mathcal{G}}^k \mathbf{1}^T)^T = \mathbf{1}^T$  where the second equality follows from the symmetry and the third from the stochasticity of  $\bar{\mathcal{G}}$ . Substituting back into equation (33), we get  $\mathbf{1}^T \bar{\mathcal{G}}_{ex}^k = \mathbf{0}^T$ . Therefore, the expression for the mean phases in equation (32) reduces to  $\bar{\varphi}[s] = \bar{\varphi}[0] + s\bar{\psi}$ . Since  $\varphi_{ex}[s] = \varphi[s] - \bar{\varphi}[s]\mathbf{1}$ , from equation (31), we get

$$\varphi_{ex}[s] = \left( \sum_{k=0}^{s-1} \bar{\mathcal{G}}_{ex}^k \right) \delta$$

Since the eigenvalues of  $\bar{\mathcal{G}}_{ex}$ , namely  $\lambda_2, \lambda_3, \dots, \lambda_N$ , are less than 1 in magnitude, the Neumann sum  $\sum_{k=0}^{\infty} \bar{\mathcal{G}}_{ex}^k$  converges to  $(\mathbb{I} - \bar{\mathcal{G}}_{ex})^{-1}$  [24]. Therefore, we have the desired result,  $\varphi_{ex}[s] \rightarrow (\mathbb{I} - \bar{\mathcal{G}}_{ex})^{-1} \delta$  as  $s \rightarrow \infty$ .

### B. Linear Programming Formulation

We begin by defining the directed phase error between nodes  $i$  and  $j$  to be  $e_{ij} = \phi_i[s] - \phi_j[s] = \mathbf{c}_{ij}^T \varphi[s]$  where  $\mathbf{c}_{ij}$  is defined as:  $\mathbf{c}_{ij}[i] = 1$ ,  $\mathbf{c}_{ij}[j] = -1$ ,  $\mathbf{c}_{ij}[l] = 0 \forall l \neq i, j$ . Since the mean component of  $\varphi[s]$  does not contribute to the error, we have,  $\mathbf{c}_{ij}^T \varphi[s] = \mathbf{c}_{ij}^T \varphi_{ex}[s] = \mathbf{c}_{ij}^T (\mathbb{I} - \bar{\mathcal{G}}_{ex})^{-1} \delta$ . We note that the phase error between nodes  $i$  and  $j$  is a linear function of the skews  $\delta$ . To obtain an estimate of the overhead required to maintain a TDM schedule, we maximize this quantity over all allowed excess frequencies  $\delta$ . The first constraint that the

excess frequencies must satisfy is  $\mathbf{1}^T \delta = 0$ . Next, we note that the excess frequencies must be bounded since the node frequencies  $F_l = f_l/f_{nom}$  are bounded by  $1 \pm \rho_{max}$ . We see this as follows: denoting the average frequency across nodes by  $\bar{\psi}$ , we have,  $F_i = \bar{\psi} + \delta_i$ . Therefore,  $\bar{\psi}$  and  $\delta_i$  must satisfy  $(1 - \rho_{max}) \leq \bar{\psi} + \delta_i \leq (1 + \rho_{max}) \forall i$ . We denote these constraints compactly as  $(1 - \rho_{max})\mathbf{1} \leq \bar{\psi}\mathbf{1} + \delta \leq (1 + \rho_{max})\mathbf{1}$ . Finally, since  $\bar{\psi}$  is the mean frequency, it is also bounded by  $1 \pm \rho_{max}$ . Therefore, the problem of maximizing the directed error between nodes  $i$  and  $j$  can be stated as,

$$\begin{aligned} \text{Maximize}_{\delta, \bar{\psi}} \quad & J_{ij} = \mathbf{c}_{ij}^T (\mathbb{I} - \bar{\mathcal{G}}_{ex})^{-1} \delta \\ \text{subject to} \quad & \delta, \bar{\psi} \in \Lambda \end{aligned}$$

where  $\Lambda$  is the set of allowed excess and mean frequencies given by  $\Lambda = \{\bar{\psi}, \delta : \mathbf{1}^T \delta = 0, (1 - \rho_{max})\mathbf{1} \leq \bar{\psi}\mathbf{1} + \delta \leq (1 + \rho_{max})\mathbf{1}, 1 - \rho_{max} \leq \bar{\psi} \leq 1 + \rho_{max}\}$ . We call this formulation problem  $\mathcal{P}_{ij}$ . This problem is a linear program since the constraints and the objective function are linear in the decision variables  $\bar{\psi}$  and  $\delta$ . To find the worst error across all neighboring nodes, we solve such a linear program for each pair of neighbors  $i$  and  $j$  and pick the largest among resulting errors.

We consider the special case of  $\bar{\psi} = 1$  to provide some insight into solutions of these linear programs. In this case, the problem of maximizing the directed phase error between nodes  $i$  and  $j$  can be simplified as,

$$\begin{aligned} \text{Maximize}_{\delta} \quad & J_{ij} = \mathbf{c}_{ij}^T (\mathbb{I} - \bar{\mathcal{G}}_{ex})^{-1} \delta \\ \text{subject to} \quad & \delta \in \tilde{\Lambda} \end{aligned}$$

where  $\tilde{\Lambda} = \{\delta : \|\delta\|_{\infty} \leq \rho_{max}, \mathbf{1}^T \delta = 0\}$ . We call this reduced problem  $\tilde{\mathcal{P}}_{ij}$ . It is known [21] that one of the optimizers of any linear program occurs at an ‘‘extreme point’’ of the feasible set (in this case,  $\tilde{\Lambda}$ ). Therefore, we characterize the extreme points of  $\tilde{\Lambda}$  to understand the structure of the solution to problem  $\tilde{\mathcal{P}}_{ij}$ . We begin with the definition of an extreme point:  $\delta_{ex}$  is an extreme point of  $\tilde{\Lambda}$ , if it cannot be expressed as  $(\delta_1 + \delta_2)/2$  for any two points  $\delta_1, \delta_2 \in \tilde{\Lambda}$  such that  $\delta_1 \neq \delta_2$ . We use this definition to make two observations that characterize the extreme points of  $\tilde{\Lambda}$ .

**Observation 1:** Let  $\delta_{ex} = (\delta_{ex,1}, \delta_{ex,2}, \dots, \delta_{ex,N})$  be an extreme point of  $\tilde{\Lambda}$ . Then, at most one of its components can take the value 0.

*Proof:* We prove the observation by contradiction. First, we note that  $\|\delta_{ex}\|_{\infty} \leq \rho_{max}$  and  $\mathbf{1}^T \delta_{ex} = 0$  since  $\delta_{ex} \in \tilde{\Lambda}$ . Assume that there are two positions  $m$  and  $m'$  such that  $\delta_{ex,m} = \delta_{ex,m'} = 0$ . We construct two vectors  $\delta_+ = \delta_{ex} + \xi$ ,  $\delta_- = \delta_{ex} - \xi$  where  $\xi = (\xi_1, \xi_2, \dots, \xi_N)$  is defined as follows:  $\xi_m = \rho_{max}/2$ ,  $\xi_{m'} = -\rho_{max}/2$ ,  $\xi_l = 0 \forall l \neq m, m'$ . We see that  $\|\delta_+\|_{\infty} \leq \max(\|\delta_{ex}\|_{\infty}, \rho_{max}/2) \leq \rho_{max}$  since,  $\|\delta_{ex}\|_{\infty} \leq \rho_{max}$ . Also,  $\mathbf{1}^T \delta_+ = \mathbf{1}^T \delta_{ex} + \mathbf{1}^T \xi = 0 + 0 = 0$ . Therefore,  $\delta_+$  is also contained in the feasible set  $\tilde{\Lambda}$ . Similarly, we can show that  $\delta_- \in \tilde{\Lambda}$ . We now arrive at a contradiction : we have just expressed the extreme point  $\delta_{ex}$  as the average of  $\delta_+$  and  $\delta_-$ , i.e.  $\delta_{ex} = (\delta_+ + \delta_-)/2$ . Therefore, we conclude that our original assumption is wrong and that at most one

component of  $\delta_{ex}$  can take the value 0.

**Observation 2:** Let  $\delta_{ex} = (\delta_{ex,1}, \delta_{ex,2}, \dots, \delta_{ex,N})$  be an extreme point of  $\tilde{\Lambda}$ . Then, each component  $\delta_{ex,i}$  takes one of the three values  $\pm\rho_{max}, 0$ .

*Proof:* We prove the observation by contradiction. First, we write  $\delta_{ex,i} = \alpha_{ex,i}\rho_{max}$  with  $|\alpha_{ex,i}| \leq 1$  since  $\|\delta_{ex}\|_\infty \leq \rho_{max}$ . Assume that for some index  $m$ ,  $\delta_{ex,m}$  takes a value other than  $\pm\rho_{max}, 0$ . Therefore, we can write  $\delta_{ex,m} = \nu\rho_{max}$  with  $-1 < \nu < 1$  and  $\nu \neq 0$ . Since  $\sum_{i=1}^N \delta_{ex,i} = 0$ , we get  $\nu = -\sum_{i \neq m} \alpha_{ex,i}$ . Since  $\nu$  is not an integer, we see that there must exist another index  $m' \neq m$  so that  $-1 < \alpha_{ex,m'} < 1$  and  $\alpha_{ex,m'} \neq 0$ . We now construct two vectors  $\delta_+ = \delta_{ex} + \xi, \delta_- = \delta_{ex} - \xi$  where  $\xi = (\xi_1, \xi_2, \dots, \xi_N)$  is defined as follows:  $\xi_m = \pi_0, \xi_{m'} = -\pi_0, \xi_l = 0 \forall l \neq m, m'$ . We choose  $\pi_0$  small enough so that  $\|\delta_+\|_\infty$  and  $\|\delta_-\|_\infty$  are both less than  $\rho_{max}$ . We can show, as before, that  $\delta_+$  and  $\delta_-$  sum to zero and thereby, conclude that  $\delta_+, \delta_- \in \tilde{\Lambda}$ . We now arrive at the same contradiction as before, and conclude that our original assumption was wrong. This shows that every entry of  $\delta_{ex}$  takes one of the three values  $\pm\rho_{max}, 0$ .

We now put these observations together to arrive at the desired result. Since the elements of  $\delta_{ex}$  add up to zero, the number of entries in  $\delta_{ex}$  that take the value  $\rho_{max}$  and  $-\rho_{max}$  must be equal. Furthermore, at most one entry in  $\delta_{ex}$  can be zero. From these facts, it is easy to see that: (1) If the number of nodes  $N$  is even, then exactly  $N/2$  entries of  $\delta_{ex}$  equal  $\rho_{max}$  and the other  $N/2$  entries equal  $-\rho_{max}$  and (2) If the number of nodes  $N$  is odd, then  $(N-1)/2$  entries of  $\delta_{ex}$  equal  $\rho_{max}$ , another  $(N-1)/2$  entries equal  $-\rho_{max}$  and there is one entry that is equal to zero. Since one of the optimizers of any linear program occurs at an extreme point, we conclude that the largest directed phase error between nodes  $i$  and  $j$  occurs with roughly half the nodes running at the maximum frequency and the other half running at the minimum frequency for *any* network topology. Furthermore, since the feasible sets for all the problems  $\mathcal{P}_{ij}$  are the same, this conclusion applies to the directed phase error between any neighboring nodes  $i$  and  $j$  and therefore, we get the desired result.

### C. Actual System - Phase Only Adjustments

We begin with the expression for the excess phases from equation (13),

$$\varphi_{ex}[s] \approx \left( \mathbb{I} + \sum_{k=1}^{s-1} G_{s-1} G_{s-2} \dots G_{s-k} \right) \delta \quad (34)$$

We denote the average of  $\varphi_{ex}[s]$  over realizations of  $\{G_t\}_{t=0}^{s-1}$ , with a fixed set of skews  $\delta$ , by  $\bar{\varphi}_{ex}[s] = \mathbb{E}(\varphi_{ex}[s])$ . Using the linearity of expectation, we get,

$$\bar{\varphi}_{ex}[s] = \left( \mathbb{I} + \sum_{k=1}^{s-1} \mathbb{E}(G_{s-1} G_{s-2} \dots G_{s-k}) \right) \delta \quad (35)$$

Since  $G_{s-1}, G_{s-2}, \dots, G_{s-k}$  are picked independent of one another, we have

$$\mathbb{E}(G_{s-1} G_{s-2} \dots G_{s-k}) = \mathbb{E}(G_{s-1}) \mathbb{E}(G_{s-2}) \dots \mathbb{E}(G_{s-k})$$

While this result is well known for scalar valued random variables, we can easily prove it for matrices as well. Since  $G_t$  is picked from  $\{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_M\}$  with probabilities  $\{p_1, p_2, \dots, p_M\}$ , we have

$$\mathbb{E}(G_t) = \sum_{i=1}^M p_i \mathcal{G}_i = \bar{\mathcal{G}} \quad \forall t$$

where  $\bar{\mathcal{G}}$  is the system matrix for the averaged system. Substituting in equation (35), we have,

$$\bar{\varphi}_{ex}[s] = \left( \sum_{k=0}^{s-1} \bar{\mathcal{G}}^k \right) \delta$$

We can now excise the first eigenmode of  $\bar{\mathcal{G}}$ , as we did in Section VIII-A, and obtain

$$\bar{\varphi}_{ex}[s] = \left( \sum_{k=0}^{s-1} \bar{\mathcal{G}}^k \right) \delta$$

The phase errors between neighbors are simply the pairwise differences between the corresponding entries of  $\varphi_{ex}[s]$ . We can obtain all such pairwise differences by operating on the excess phases  $\varphi_{ex}[s]$  with a matrix  $C$ . We denote the maximum error between any pair of nodes by  $\|C\varphi_{ex}[s]\|_\infty$ . We can easily show that  $\|v\|_\infty$  is a convex function of its argument  $v$ . Using this fact, we apply Jensen's inequality to  $\|C\varphi_{ex}[s]\|_\infty$  and obtain,

$$\begin{aligned} \mathbb{E}(\|C\varphi_{ex}[s]\|_\infty) &\geq \|\mathbb{E}(C\varphi_{ex}[s])\|_\infty \\ &= \|C\mathbb{E}(\varphi_{ex}[s])\|_\infty = \|C\bar{\varphi}_{ex}[s]\|_\infty \end{aligned} \quad (36)$$

For a given set of skews  $\delta$ , we recognize that  $\|C\bar{\varphi}_{ex}[s]\|_\infty$  is the maximum pairwise phase difference for the averaged system and  $\mathbb{E}(\|C\varphi_{ex}[s]\|_\infty)$  is the average of the maximum pairwise phase difference in the original system. Thus, we conclude that the largest error from the averaged system provides a lower bound to the worst error in the actual system, averaged over many realizations of communication patterns.

### D. Estimating skews from raw phases

We show here that  $(\mathbb{I} - \bar{\mathcal{G}}) \varphi[s] = (\mathbb{I} - \bar{\mathcal{G}}_{ex}) \varphi_{ex}[s]$ . First, since  $\mathbf{1}^T \varphi_{ex} = 0$ , we have,

$$\begin{aligned} (\mathbb{I} - \bar{\mathcal{G}}_{ex}) \varphi_{ex}[s] &= \left( \mathbb{I} - \bar{\mathcal{G}}_{ex} - \frac{\mathbf{1}\mathbf{1}^T}{N} \right) \varphi_{ex}[s] \\ &= (\mathbb{I} - \bar{\mathcal{G}}) \varphi_{ex}[s] \end{aligned} \quad (37)$$

where the second equality follows from the fact that  $\bar{\mathcal{G}} = \bar{\mathcal{G}}_{ex} + \mathbf{1}\mathbf{1}^T/N$ . Next, since  $\bar{\mathcal{G}}$  is stochastic, we have,  $(\mathbb{I} - \bar{\mathcal{G}}) \mathbf{1} = \mathbf{1} - \mathbf{1} = \mathbf{0}$ . Therefore, we can add  $(\mathbb{I} - \bar{\mathcal{G}}) \bar{\varphi}[s] \mathbf{1}$  to the right hand side of equation 37 and obtain,

$$\begin{aligned} (\mathbb{I} - \bar{\mathcal{G}}_{ex}) \varphi_{ex}[s] &= (\mathbb{I} - \bar{\mathcal{G}}) (\varphi_{ex}[s] + \bar{\varphi}[s] \mathbf{1}) \\ &= (\mathbb{I} - \bar{\mathcal{G}}) \varphi[s] \end{aligned} \quad (38)$$

where the second equality follows from the fact that  $\varphi[s] = \bar{\varphi}[s] \mathbf{1} + \varphi_{ex}[s]$ . Thus, we have the desired result.

### E. Expression for the skew estimate - Averaged system

We first obtain an expression for the entries of  $\bar{\mathcal{G}}$ . By definition, the  $(i, j)^{th}$  element of  $\bar{\mathcal{G}}$  is  $\bar{\mathcal{G}}(i, j) = \sum_{m=1}^M p_m \mathcal{G}_m(i, j)$ . We consider the case  $i \neq j$  first. If  $\mathcal{N}_j$  transmits a packet to  $\mathcal{N}_i$  when the  $m^{th}$  matching matrix  $\mathcal{G}_m$  is in operation, we have  $\mathcal{G}_m(i, j) = \beta$ ; otherwise,  $\mathcal{G}_m(i, j) = 0$ . Therefore, the only matrices  $\mathcal{G}_m$  that contribute to  $\bar{\mathcal{G}}(i, j)$  are those such that  $\mathcal{G}_m(i, j) = \beta$ . Summing over the probabilities of picking such matrices, we can obtain the probability that  $\mathcal{N}_j$  transmits to  $\mathcal{N}_i$ . We denote this probability by  $q_{j \rightarrow i}$ . With these definitions, we see that  $\bar{\mathcal{G}}(i, j)$  is exactly  $q_{j \rightarrow i} \beta$ . Since  $\bar{\mathcal{G}}$  is stochastic, we have  $\bar{\mathcal{G}}(i, i) = 1 - \beta \sum_{j \neq i} q_{j \rightarrow i}$ . We denote the  $i^{th}$  component of  $\varphi_\infty$  by  $\varphi_{\infty, i}$  and that of  $\bar{\mathcal{L}}\varphi_\infty$  by  $[\bar{\mathcal{L}}\varphi_\infty]_i$ . We now expand the matrix vector product  $\bar{\mathcal{L}} = (\mathbb{I} - \bar{\mathcal{G}})\varphi_\infty$  and equate its  $i^{th}$  component to  $\delta_i$ . By this process, we get,

$$[\bar{\mathcal{L}}\varphi_\infty]_i = \beta \sum_{j=1}^N q_{j \rightarrow i} (\varphi_{\infty, i} - \varphi_{\infty, j}) = \delta_i \quad (39)$$

### F. Evolution of the excess phases across a round

Applying equation (5) to phase evolution within a round, from slot  $(s-1, r)$  to  $(s, r)$ , we have

$$\varphi[s, r] = \bar{\mathcal{G}}\varphi[s-1, r] + \delta[s, r] + \bar{\psi}[s, r]\mathbf{1} \quad (40)$$

Since the frequencies are unchanged from the beginning of round  $r$ , the mean and excess frequencies are also unchanged from their values at the start of round  $r$  i.e.  $\bar{\psi}[s, r] = \bar{\psi}[0, r]$  and  $\delta[s, r] = \delta[0, r]$ . Using this observation, and stepping back in time repeatedly using equation (5), we obtain

$$\varphi[s, r] = s\bar{\psi}[0, r]\mathbf{1} + \bar{\mathcal{G}}^s \varphi[0, r] + \left( \sum_{k=0}^{s-1} \bar{\mathcal{G}}^k \right) \delta[0, r] \quad (41)$$

We now compute the mean phase across nodes in slot  $s$  of round  $r$ , denoted by  $\bar{\varphi}[s, r]$ . Since  $\bar{\varphi}[s, r] = \mathbf{1}^T \varphi[s, r]/N$ , we have,

$$\bar{\varphi}[s, r] = s\bar{\psi}[0, r] \frac{\mathbf{1}^T \mathbf{1}}{N} + \mathbf{1}^T \bar{\mathcal{G}}^s \varphi[0, r] + \mathbf{1}^T \left( \sum_{k=0}^{s-1} \bar{\mathcal{G}}^k \right) \delta[0, r] \quad (42)$$

Consider the terms separately. The term  $\mathbf{1}^T \mathbf{1}/N$  is simply 1. We see that  $\mathbf{1}^T \bar{\mathcal{G}}^k = \mathbf{1}^T \forall k$ :  $\mathbf{1}^T \bar{\mathcal{G}}^k = ((\bar{\mathcal{G}}^k)^T \mathbf{1})^T = (\bar{\mathcal{G}}^k \mathbf{1})^T = \mathbf{1}^T$  where the second equality follows from the symmetry and the third from the stochasticity of  $\bar{\mathcal{G}}$ . Thus, the second term in equation (42) simplifies to  $\mathbf{1}^T \varphi[0, r] = \bar{\varphi}[0, r]$ . We also use the fact that  $\mathbf{1}^T \bar{\mathcal{G}}^k = \mathbf{1}^T$  to show that the third term  $\sum_{k=0}^{s-1} (\mathbf{1}^T \bar{\mathcal{G}}^k) \delta$  is annihilated:  $\sum_{k=0}^{s-1} (\mathbf{1}^T \bar{\mathcal{G}}^k) \delta = \sum_{k=0}^{s-1} \mathbf{1}^T \delta = 0$  where the second equality follows from the fact that  $\mathbf{1}^T \delta = 0$ . Putting these terms together, the mean phase in slot of  $s$  of round  $r$  is given by  $\bar{\varphi}[s, r] = s\bar{\psi}[0, r] + \bar{\varphi}[0, r]$ . Since the excess phases  $\varphi_{ex}[s, r]$  are given by  $\varphi[s, r] - \bar{\varphi}[s, r]\mathbf{1}$ , we have, from equation (41),

$$\varphi_{ex}[s, r] = \bar{\mathcal{G}}^s \varphi[0, r] - \bar{\varphi}[0, r]\mathbf{1} + \sum_{k=0}^{s-1} \bar{\mathcal{G}}^k \delta \quad (43)$$

Since  $\bar{\mathcal{G}}^s$  is also stochastic, we have  $\bar{\mathcal{G}}^s \mathbf{1} = \mathbf{1}$ , giving us,

$$\varphi_{ex}[s, r] = \bar{\mathcal{G}}^s (\varphi[0, r] - \bar{\varphi}[0, r]\mathbf{1}) + \sum_{k=0}^{s-1} \bar{\mathcal{G}}^k \delta \quad (44)$$

$$= \bar{\mathcal{G}}^s \varphi_{ex}[0, r] + \sum_{k=0}^{s-1} \bar{\mathcal{G}}^k \delta \quad (45)$$

where the second equality follows from the definition of the excess phases. Since  $\bar{\mathcal{G}}^k \delta = \bar{\mathcal{G}}_{ex}^k \delta$  (see Section VIII-A), we obtain the desired result:

$$\varphi_{ex}[s, r] = \bar{\mathcal{G}}^s \varphi_{ex}[0, r] + \sum_{k=0}^{s-1} \bar{\mathcal{G}}_{ex}^k \delta$$

### G. Recursive bounds on the excess phases

The phases in the last slot of round  $r$  and the first slot of round  $r+1$  are related by the phase evolution equation:  $\varphi[0, r+1] = \varphi[W_r - 1, r] + \bar{\psi}[0, r]\mathbf{1} + \delta[0, r]$ . We can once again calculate and subtract out the mean from both sides of this equation to infer that  $\varphi_{ex}[0, r+1] = \varphi_{ex}[W_r - 1, r] + \delta[0, r]$ . Taking infinity norms on both sides and applying the triangle inequality, we get

$$\|\varphi_{ex}[0, r+1]\|_\infty \leq \|\varphi_{ex}[W_r - 1, r]\|_\infty + \|\delta[0, r]\|_\infty \quad (46)$$

However, we know from equation (17) that,

$$\varphi_{ex}[W_r - 1, r] = \bar{\mathcal{G}}_{ex}^{W_r - 1} \varphi_{ex}[0, r] + D_{W_r - 1} \delta[0, r] \quad (47)$$

Taking infinity norms on both sides of this equation and applying the triangle inequality, we get,

$$\|\varphi_{ex}[W_r - 1, r]\|_\infty \leq \|\bar{\mathcal{G}}_{ex}^{W_r - 1}\|_\infty \|\varphi_{ex}[0, r]\|_\infty + \|D_{W_r - 1}\|_\infty \|\delta[0, r]\|_\infty \quad (48)$$

Substituting this in equation (46), we get,

$$\|\varphi_{ex}[0, r+1]\|_\infty \leq \|\bar{\mathcal{G}}_{ex}^{W_r - 1}\|_\infty \|\varphi_{ex}[0, r]\|_\infty + (1 + \|D_{W_r - 1}\|_\infty) \|\delta[0, r]\|_\infty \quad (49)$$

which proves the desired result.

### H. Proof of Proposition 4

*Proof:* As usual, we only prove the statement for the case where  $S_1\{r\} \neq \emptyset$ . Let  $\mathcal{N}^\#$  be the node with the maximum frequency in round  $r$ . Since  $S_1\{r\} \neq \emptyset$ ,  $\mathcal{N}^\# \in S_1\{r\}$ . Thus, from Proposition (1),  $\mathcal{N}^\#$  reduces its frequency after round  $r$ . Therefore,  $F_{\mathcal{N}^\#}\{r+1\} = F_{\mathcal{N}^\#}\{r\} - \mu$ . We now prove the proposition by considering three cases.

*Case 1:* Let  $\mathcal{N}_i$  be any node other than  $\mathcal{N}^\#$  in  $S_1\{r\} \cup S_2\{r\}$ . From Proposition (1),  $\mathcal{N}_i$  also reduces its frequency after round  $r$ , or  $F_i\{r+1\} = F_i\{r\} - \mu$ . Therefore,  $\mathcal{N}_i$  continues to have a frequency smaller or equal to that of  $\mathcal{N}^\#$  in round  $r+1$ :  $F_{\mathcal{N}^\#}\{r+1\} - F_i\{r+1\} = (F_{\mathcal{N}^\#}\{r\} - \mu) - (F_i\{r\} - \mu) = F_{\mathcal{N}^\#}\{r\} - F_i\{r\}$ , which, by definition, is greater than or equal to zero.

*Case 2:* Let  $\mathcal{N}_i$  be a node in  $S_3\{r\}$  or  $S_4\{r\}$  or  $S_5\{r\}$ . Thus,  $\mathcal{N}_i$ 's frequency in the  $r^{th}$  round is not too large:

$F_i\{r\} \leq \bar{\psi}\{r\} + \epsilon + \chi$ . Furthermore, from Proposition (1),  $\mathcal{N}_i$  does not increase its frequency at the end of round  $r$  i.e.  $F_i\{r+1\} \leq F_i\{r\}$ . Therefore, we have,  $F_i\{r+1\} \leq \bar{\psi}\{r\} + \epsilon + \chi$ . However,  $\mathcal{N}^\#$ 's frequency in round  $r+1$  is larger than this:  $F_{\mathcal{N}^\#}\{r+1\} = F_{\mathcal{N}^\#}\{r\} - \mu > (\bar{\psi}\{r\} + \epsilon + \mu + \chi) - \chi = \bar{\psi}\{r\} + \epsilon + \chi$  (the inequality follows from the fact that  $\mathcal{N}^\# \in S_1\{r\}$ ).

*Case 3:* Consider  $\mathcal{N}_i$  with a negative excess frequency in round  $r$ . From Proposition (2), we know that  $F_i\{r+1\} < \bar{\psi}\{r\}$ . However, we have already shown that  $F_{\mathcal{N}^\#}\{r+1\} > \bar{\psi}\{r\} + \epsilon + \chi$ . Therefore,  $F_{\mathcal{N}^\#}\{r+1\} > \bar{\psi}\{r\} > F_i\{r+1\}$ .

Therefore,  $F_{\mathcal{N}^\#}\{r+1\} \geq F_i\{r+1\} \forall i$  and  $\mathcal{N}^\#$  continues to be the node with the highest frequency in round  $r+1$ . Since it was also the node with maximum frequency in the  $r^{\text{th}}$  round and its frequency decreased by  $\mu$  at the end of round  $r$ , we have  $F_{\max}\{r+1\} = F_{\max}\{r\} - \mu$ . ■

### I. Proof of Proposition 6

*Proof:* We only prove the statement for nodes with positive excess frequencies in the  $r^{\text{th}}$  round. For notational ease, we denote the change in the mean frequency from round  $r$  to round  $r+1$  by  $\Delta\bar{\psi}\{r\} = \bar{\psi}\{r+1\} - \bar{\psi}\{r\}$ . From corollary 1, we have  $|\Delta\bar{\psi}| \leq \mu$ .

*Case 1:* Let  $\mathcal{N}_i \in S_2\{r\}$ . From Proposition 1, we know that  $\mathcal{N}_i$  decreases its frequency at the end of round  $r$ , or  $F_i\{r+1\} = F_i\{r\} - \mu$ . Consequently, its excess frequency in round  $r+1$  is,  $\delta_i\{r+1\} = F_i\{r+1\} - \bar{\psi}\{r+1\} = F_i\{r\} - \mu - \bar{\psi}\{r+1\}$ . Adding and subtracting  $\bar{\psi}\{r\}$  to the extreme right hand side and grouping terms, we get  $\delta_i\{r+1\} = \delta_i\{r\} - \mu - \Delta\bar{\psi}\{r\}$ . Since  $\mathcal{N}_i \in S_2\{r\}$ , we have bounds on its excess frequency,  $\epsilon + \chi < \delta_i\{r\} \leq \epsilon + \chi + \mu$ . Using these bounds and the fact that  $|\Delta\bar{\psi}\{r\}| \leq \mu$  in the expression for  $\delta_i\{r+1\}$ , we get,  $\epsilon + \chi - 2\mu \leq \delta_i\{r+1\} \leq \epsilon + \chi + \mu$ . We also have  $\epsilon + \chi - 2\mu = -(\epsilon + \mu + \chi) + (2\epsilon - \mu + 2\chi) > -(\epsilon + \mu + \chi)$  since we have chosen  $\epsilon > \mu + \chi$ . Therefore,  $|\delta_i\{r+1\}| \leq (\epsilon + \chi + \mu)$ .

*Case 2:* Let  $\mathcal{N}_i$  be in  $S_3\{r\} \cup S_4\{r\} \cup S_5\{r\}$ . From Proposition (1), we know that  $\mathcal{N}_i$  does not increase its frequency after round  $r$ , or  $F_i\{r+1\} \leq F_i\{r\}$ . From Proposition 2, we also know that it does not decrease its frequency too much,  $F_i\{r+1\} > \bar{\psi}\{r\}$ . Chaining these inequalities, we get,  $\bar{\psi}\{r\} \leq F_i\{r+1\} \leq F_i\{r\}$ . We first subtract  $\bar{\psi}\{r\}$  throughout and then, add and subtract  $\bar{\psi}\{r+1\}$  to the middle term and obtain,  $0 \leq \delta_i\{r+1\} + \Delta\bar{\psi}\{r\} \leq \delta_i\{r\}$ . Equivalently, we have  $-\Delta\bar{\psi}\{r\} \leq \delta_i\{r+1\} \leq \delta_i\{r\} - \Delta\bar{\psi}\{r\}$ . Since  $\mathcal{N}_i \in S_3\{r\} \cup S_4\{r\} \cup S_5\{r\}$ , we have  $0 \leq \delta_i\{r\} \leq \epsilon + \chi$ . Using this bound and the fact that  $|\Delta\bar{\psi}\{r\}| \leq \mu$  in the expression for  $\delta_i\{r+1\}$  we get,  $-\mu \leq \delta_i\{r+1\} \leq \epsilon + \mu + \chi$ . Since  $-\mu > -(\epsilon + \mu + \chi)$ , we conclude that  $|\delta_i\{r+1\}| \leq (\epsilon + \mu + \chi)$ . ■

### J. LLN arguments for the actual system

Since the third term in equation (23) is sufficient to obtain the excess-averaged-phase we denote it by  $\varphi_{av,suf}$ . We set

$\varphi_{av,suf} = H\delta[0]$  where

$$H \triangleq \frac{1}{S_R} \sum_{s'=2}^{S_R-1} \left[ \mathbb{I} + \sum_{p=1}^{s'-1} G_{s'-1} G_{s'-2} \dots G_{s'-p} \right] \\ = \frac{S_R-2}{S_R} \mathbb{I} + \frac{1}{S_R} \sum_{s'=2}^{S_R-1} \sum_{p=1}^{s'-1} G_{s'-1} G_{s'-2} \dots G_{s'-p} \quad (50)$$

We now express the matrix  $H$  as the sum of matrices  $H_p$ ,  $0 \leq p \leq S_R - 2$ , with the matrix  $H_p$  collecting all the terms in  $H$  which are products of  $p$  system matrices. To do this, we interchange the order of the sums in equation (50) and get,

$$H = \frac{S_R-2}{S_R} \mathbb{I} + \frac{1}{S_R} \sum_{p=1}^{S_R-2} \sum_{s'=p+1}^{S_R-1} G_{s'-1} G_{s'-2} \dots G_{s'-p}$$

We can set  $H_0 = (S_R - 2)/(S_R)\mathbb{I}$  and  $H_p = \frac{1}{S_R} \sum_{s'=p+1}^{S_R-1} G_{s'-1} G_{s'-2} \dots G_{s'-p}$ ,  $1 \leq p \leq S_R - 2$  and see that each term in the sum for  $H_p$  is a product of  $p$  system matrices. We use the LLN, and some approximations, to show that  $H_p$  is virtually independent of the precise schedule in operation during this round of slots.

**An expression for  $H_p$  for small values of  $p$ :** We begin with the simplest of these matrices  $H_0$ . For large values of  $S_R$ , we can approximate  $(S_R-2)/(S_R)$  by 1 and consequently  $H_0 \approx \mathbb{I}$  when  $S_R \gg 2$ . Next, we consider the simplest “nontrivial” matrix  $H_1$ . By definition, we have,

$$H_1 = \frac{1}{S_R} \sum_{s'=2}^{S_R-1} G_{s'-1} = \frac{G_1 + G_2 + \dots + G_{S_R-2}}{S_R} \quad (51)$$

Since the matrices  $\{G_t\}_{t=0}^\infty$  are chosen independently from the set  $\mathcal{S}_m = \{G_1, G_2, \dots, G_M\}$  with probabilities  $\{p_1, p_2, \dots, p_M\}$  respectively, we can use the law of large numbers when  $S_R$  is large and obtain,

$$H_1 = \frac{S_R-2}{S_R} \frac{G_1 + G_2 + \dots + G_{S_R-2}}{S_R-2} \approx \frac{S_R-2}{S_R} \bar{G} \quad (52)$$

where  $\bar{G}$ , as before, denotes the average system matrix  $\sum_{i=1}^M p_i G_i$ . When  $S_R \gg 2$ , we can approximate  $(S_R-2)/S_R$  by 1 and therefore,  $H_1 \approx \bar{G}$ .

We now consider the matrix  $H_2$ . By definition,  $H_2 = \frac{1}{S_R} \sum_{s'=3}^{S_R-1} G_{s'-1} G_{s'-2}$ . Expanding the sum, we get,

$$H_2 = \frac{G_2 G_1 + G_3 G_2 + G_4 G_3 + \dots + G_{S_R-2} G_{S_R-3}}{S_R} \quad (53)$$

We cannot apply the law of large numbers to equation (53) directly because the terms in the sum are not independent. For example, the first two terms have  $G_2$  in common and therefore, cannot be independent. However, the correlation between the terms in the sum is fairly weak - only adjacent terms have any dependence at all. For example, the first term  $G_2 G_1$  is completely independent of the third term  $G_4 G_3$  and any future terms. Therefore, we can break the dependence among the terms by splitting the sum into two parts: one part consisting

of the odd terms and the other part containing the even terms. After such a split, we get,

$$H_2 = \frac{G_2 G_1 + G_4 G_3 + \dots G_{S_R-3} G_{S_R-4}}{S_R} + \frac{G_3 G_2 + G_5 G_4 + \dots G_{S_R-2} G_{S_R-3}}{S_R} \quad (54)$$

where we have assumed, for concreteness, that  $S_R$  is odd. Since the constituent elements of the doublet matrices, such as  $G_2 G_1, G_4 G_3$  are independent, each doublet has an average value of  $\bar{\mathcal{G}}^2$ . Now, consider either sum in equation (54): it consists of  $(S_R - 3)/2$  terms, which are all independent of one another. Applying the law of large numbers separately to each sum, we see that they both approach  $\frac{S_R-3}{2S_R} \bar{\mathcal{G}}^2$  when  $S_R$  is large. Therefore,  $H_2 \approx 2 \times \frac{S_R-3}{2S_R} \bar{\mathcal{G}}^2 \approx \bar{\mathcal{G}}^2$  for large values of  $S_R$ . In a completely analogous fashion, we can split the expression for  $H_p$  into  $p$  ‘‘interleaved’’ sums and use the LLN on each sum to conclude that  $H_p \approx \frac{S_R-(p+1)}{S_R} \bar{\mathcal{G}}^p \approx \bar{\mathcal{G}}^p$  as long as  $S_R - (p + 1)$  is large or equivalently,  $S_R \gg p + 1$ . Next, we tackle the problem of simplifying  $H_p$  when  $p$  becomes ‘‘large’’.

**An expression for  $H_p$  for large values of  $p$ :** Each term in the sum for  $H_p = \frac{1}{S_R} \sum_{s'=p+1}^{S_R-1} G_{s'-1} G_{s'-2} \dots G_{s'-p}$  is a product of  $p$  stochastic matrices. Thus, if  $p$  is larger than the critical limit  $S_w$ , a generic term of the form  $G_{s'-1} G_{s'-2} \dots G_{s'-p}$ , is equal to a rank-one matrix  $\mathbf{1} \alpha_{s',p}^T$ . Substituting into the expression for  $H_p$ , we get

$$H_p = \mathbf{1} \alpha_p^T \quad (55)$$

where  $\alpha_p = \sum_{s'=p+1}^{S_R-1} \alpha_{s',p} / (S_R)$  when  $p > S_w$ .

We put these results together in the form of three conclusions. Since we have chosen the slots  $S_R$  such that it is much larger than both  $S_{\bar{\mathcal{G}}}$  and  $S_w$  (i.e  $S_R \gg S^* = \max(S_{\bar{\mathcal{G}}}, S_w)$ ), we have:

- 1) Since  $S^*$  is much smaller than  $S_R$ ,  $H_p \approx \bar{\mathcal{G}}^p$ ,  $p < S^*$
- 2) Since  $S^*$  is larger than the critical limit  $S_w$ ,  $H_p \approx \mathbf{1} \alpha_p^T \forall p \geq S^*$  for some vector  $\alpha_p$
- 3) Since  $S^*$  is larger than  $S_{\bar{\mathcal{G}}}$ ,  $\lambda_2^{S^*}$  is negligible

Adding these terms to form  $H$  and substituting into the expression for  $\varphi_{av,suf}$ , we get,

$$\varphi_{av,suf} = H \delta[0] \approx \kappa \mathbf{1} + \sum_{p=0}^{S^*-1} \bar{\mathcal{G}}^p \delta[0] \quad (56)$$

where  $\kappa = \sum_{p=S^*}^{S_R-2} (\alpha_p^T \delta[0])$ . Since the first term only contributes to the mean of the averaged phases, it can be thrown out. Thus, we get the *excess-averaged-phases*  $\varphi_{av,ex}$  to be

$$\varphi_{av,ex} \approx \sum_{p=0}^{S^*-1} \bar{\mathcal{G}}^p \delta[0]$$

Since  $\bar{\mathcal{G}}^p \delta[0] = \bar{\mathcal{G}}_{ex}^p \delta[0]$  (see Section VIII-A), we have,

$$\varphi_{av,ex} \approx \sum_{p=0}^{S^*-1} \bar{\mathcal{G}}_{ex}^p \delta[0] \quad (57)$$

We can now write  $\sum_{p=0}^{S^*-1} \bar{\mathcal{G}}_{ex}^p = \sum_{p=0}^{\infty} \bar{\mathcal{G}}_{ex}^p - \sum_{p=S^*}^{\infty} \bar{\mathcal{G}}_{ex}^p$ . Since all eigenvalues of  $\bar{\mathcal{G}}_{ex}$  are less than 1, the Neumann sum  $\sum_{p=0}^{\infty} \bar{\mathcal{G}}_{ex}^p$  converges to  $(\mathbb{I} - \bar{\mathcal{G}}_{ex})^{-1}$ . Approximating  $\bar{\mathcal{G}}_{ex}^s$  by the dominant term in its spectral decomposition,  $\lambda_2^s \mathbf{v}_2 \mathbf{v}_2^T$ , we get,

$$\sum_{p=S^*}^{\infty} \bar{\mathcal{G}}_{ex}^p = \frac{\lambda_2^{S^*}}{1 - \lambda_2} \mathbf{v}_2 \mathbf{v}_2^T \approx 0$$

where the second equality follows from the fact that  $\lambda_2^{S^*}$  is negligible. Using these conclusions in equation (57), we get the desired result,

$$\varphi_{av,ex} \approx (\mathbb{I} - \bar{\mathcal{G}}_{ex})^{-1} \delta[0]$$