Cycle Synchronization in Cyclone Networks

Doug Szajda, Simon Hawkin and Ashok Agrawala

Institute for Advanced Computer Studies and Department of Computer Science
University of Maryland
College Park, MD 20742

***Abstract*— Networks with time-cyclic allocation of node resources (e.g. SONET or a wireless network using TDMA media access scheduling) require a degree of synchronization in order to guarantee that cycle lengths and drifts remain within acceptable tolerances. Cyclone is a class of such networks. By requiring applications to reserve resources in both time and space, a balance is achieved between arriving and departing flows, ensuring a lack of congestion and eliminating the need for flow control. This paper presents a lightweight cycle synchronization algorithm for Cyclone networks, and shows through both theoretical analysis and simulation that even in the presence of timing errors, both cycle length jitter and cycle phase differences fall well within tolerable levels.**

***Keywords*— Cyclone, synchronization, distributed algorithms, performance evaluation, modeling.**

1 Introduction

A Cyclone network is a class of connection oriented synchronous communication network that utilizes a cyclic approach to node resource management [Lee98]. Resources are reserved for the duration of a *cycle*, a fixed length period during which fixed size *chunks* of data are transmitted. Because reservations specify both time (i.e. cycle number) and link (i.e. the link from node i to node j), there is no run-time contention for resources between different scheduled communication tasks. This allows the system to provide deterministic (predictable) service levels, and results in zero delay jitter, lossless data transmission, and efficient processing of datagrams on every node of the transmission path.

Nodes in a Cyclone network (called *cyclonodes*) are controlled by clocks local to the individual nodes. Because discrepancies among the local clocks can be expected, and because Cyclone network protocols require tight temporal coordination, synchronization between nodes is essential. For this purpose, each cycle is divided into a *transmission period* of fixed (and global) duration and an *idle period* of variable duration. The variability of the length of the idle period is used to adjust the lengths of the cycles on individual nodes, ensuring both relatively uniform cycle lengths and very low phase differences.

Relationships between individual timelines in a distributed system have been studied since the classic work by Lamport [Lam78]. Mills [Mil92] developed the NTP protocol which is the basis for time synchronization on the Internet (see also [Mil] for an extensive list of related literature).

Department of Computer Science, University of Maryland, College Park, MD 20742

In this context, a fundamental issue is avoiding discrepancies between local timelines, typically through synchronizing clock values. That the algorithm presented here does not achieve clock synchronization, but instead achieves the necessary *cycle* synchronization. That is, the local clocks can, over time, have widely varying time readings—the times they provide are not in any way synchronized with each other or with global time, and local clocks do not adjust their time readings in any way. Rather, it is the cycle start and end times that remain synchronized.

This paper presents the cycle synchronization algorithm developed for Cyclone networks, along with an analysis of relevant convergence properties. Although developed for Cyclone, the algorithm is applicable, with slight modifications, to other synchronous network models, such as SONET [Gor97], or for implementing a flexible TDMA scheduling of nodes in a mobile ad-hoc network. The details of the changes required for SONET will be described in a subsequent article.

In Section 2 we present the details of the network model under consideration, and discuss the interplay between local and global clock times in this model. Section 3 covers the details of the algorithm, including minor modifications required to handle some practical implementation details. Section 4 presents an analysis of algorithm convergence properties, first under ideal conditions, and then in the context of the more realistic error models that can be expected in a practical implementation. Section 5 covers our simulation results. Our conclusions are presented in Section 6.

2 The Network Model

Cyclone nodes (also called *cyclonodes*) are connected with point-to-point full duplex links. Operations on any particular node are controlled by a clock local to that node. We assume that all links operate continuously, and that a node can record the arrival time of any bit, as reported by its local clock. We assume also that the graph that represents the Cyclone network topology is connected. Network topology naturally introduces the concept of neighbors. We call node j a *neighbor* of node i, and write $i \leftrightarrow j$, if there is a link between nodes i and j. The neighborhood of node i is defined as the set

$$U_i \equiv \{j : i \leftrightarrow j\} \cup \{i\}.$$

That is, we include i itself in the set of neighbors of i. Finally, we denote the latency of the j to i direction of the link $i \leftrightarrow j$ by ℓ_{ij}. Note that while all links are bidirectional, it need not be the case that $\ell_{ij} = \ell_{ji}$.

2.1 The Clock Model and Notational Conventions

Much of the analysis in the sequel involves differences between global time and the local time readings at various nodes, as well as relations between the local readings at neighboring nodes. Specifically, we will be interested in knowing the start and finish times for each cycle according to the clocks local to each node, according to the clocks of neighboring nodes, and according to global time. As an example, we may wish to know the start time of cycle k on node j, as interpreted by the clock on node i. We denote this by $s_j^i(k)$. That is, the superscript is the index of the clock according to which the time is interpreted, the subscript is the index of the node on which the cycle occurred, and the "argument" provides the index of the specific cycle. When providing generic time readings, a similar notation is used, with the added proviso that times with no superscript indicate global time.

Every cyclonode operates in cycles, which as mentioned earlier consists of a data transmission period followed by an idle period. The length of the data transmission period on node i is equal to the global constant T, as measured by clock i. Without synchronization, the length of the idle period would be the global constant Y, in which case we would have $C = T + Y$. Because every node can record the arrival time of any bit on its input links, information about the length of the transmission periods of any neighbor can be collected by observing the start and finish times of the data transmissions received from that neighbor, modulo delays due to link latencies and discretization errors.

Our first order model of the local clocks assumes a constant drift rate. Specifically, we assume that the local time t^i as given by clock i is linearly related to the global time t according to the relation

$$t^i = r_i t + \beta_i,$$

where both the clock rate r_i and phase shift β_i are fixed. We assume also, for the moment, that link latencies are fixed as well. These assumptions, which allow us to provide a static analysis of the behavior of the synchronization algorithm, will be relaxed following the initial analysis.

2.2 Global and Local Time

Consider an interval of time Δt between two events that occurred at global times t_1 and t_2 respectively. According to the global time frame, the length of this time interval is $\Delta t = t_2 - t_1$. According to clock i, however, these events occurred at times $t_1^i = r_i t_1 + \beta_i$ and $t_2^i = r_i t_2 + \beta_i$ respectively. The time interval Δt^i between these events is calculated, according to clock i, as

$$\Delta t^i = t_2^i - t_1^i = r_i(t_2 - t_1) = r_i \Delta t.$$

This same interval of time would be recorded on node j as having duration $\Delta t^j = r_j \Delta t$. Combining these two relations, we see that the values of an interval of time on nodes i and j are related by

$$\Delta t^j = r_j \Delta t = r_j \left(\frac{\Delta t^i}{r_i} \right) = \left(\frac{r_j}{r_i} \right) \Delta t^i, \tag{1}$$

where $\frac{r_j}{r_i}$ is a constant *rate ratio.* Equation (1) has the following interpretation: when two events occur at a distance of Δt^i units of time according to clock i, these events occur at a distance of $\frac{r_j}{r_i} \Delta t^i$ according to clock j.

3 The Algorithm

The working principle of the algorithm is straightforward:

Any node can estimate the clock rate of any neighbor by observing the start and end times of the data transmission period of that neighbor, knowing that the neighbor believes that it has transmitted for a time interval of duration T. Using this clock rate information, along with the start time of the neighbor's current cycle, the node can project the time (as measured by its own clock) at which that neighbor will begin the next cycle. The cycle synchronization algorithm requires that the node begin its next cycle at the time obtained by averaging these projected next cycle start times for all of its neighbors (including itself).

Since each node is running the same algorithm, cycle start times will synchronize, with the rate of synchronization depending on the degree of connectivity among nodes. In the case of a complete network topology graph (and with our simplified initial assumptions), convergence occurs after only one cycle.

Consider adjacent nodes i and j. Because the length of each data transmission period on node j is T as measured by clock j, this same time interval must be $T_i^j \equiv \frac{r_i}{r_j} T$ as measured at node i. Thus, if node i is able to measure the length of the data transmission period of node j, it can compute the pairwise rate ratio:

$$\frac{r_i}{r_j} = \frac{T_j^i}{T}.$$

Now, let $s_j(k)$ and $f_j(k)$ be the global starting and finishing times of the k-th data transmission period on node j. Node i will observe these events at times $s_j^i(k) \equiv s_j(k) + \ell_{ij}$ and $f_j^i(k) \equiv f_j(k) + \ell_{ij}$ respectively according to its local clock. Node i can therefore measure T_j^i for every node j in its neighborhood as

$$T_j^i = f_j^i(k) - s_j^i(k),$$

and can thus calculate $\frac{r_i}{r_j}$ as the fraction $\frac{T_j^i}{T}$. Believing that node j began the current cycle at time $s_j^i(k)$, and knowing that the global cycle length is C, node i thus projects that

j will start the next cycle at time

$$s_j^i(k) + \frac{r_i}{r_j}C,$$

which can be expressed as a function of the data transmission times received from j as

$$s_j^i(k) + \frac{f_j^i(k) - s_j^i(k)}{T}C.$$

Thus, when averaging the projected start times for its neighbors (including itself), the start time of cycle $k+1$ according to clock i, $s_i^i(k+1)$, is given by

$$\begin{aligned} s_i^i(k+1) &= \frac{1}{|U_i|}\sum_j \left(s_j^i(k) + \frac{f_j^i(k) - s_j^i(k)}{T}C \right) \\ &= \frac{1}{|U_i|}\sum_j \left(s_j^i(k) + \frac{r_i}{r_j}C \right), \end{aligned}$$

which may be expressed in global time by

$$s_i(k+1) = \frac{1}{|U_i|}\sum_j \left(s_j(k) + \ell_{ij} + \frac{C}{r_j} \right), \qquad (2)$$

where in each of the above equations, $|U_i|$ denotes the cardinality of the set U_i.

In summary, each node sets the start time (according to its own clock) of the subsequent cycle at

$$\frac{1}{|U_i|}\sum_j \left(s_j^i(k) + \frac{f_j^i(k) - s_j^i(k)}{T}C \right). \qquad (3)$$

Note that because of the simplicity of the algorithm, it can be implemented either in software or hardware. Moreover, memory requirements are minimal—the node maintains the values T and C, and repeatedly measures the real-valued parameter s_j^i and f_j^i for each neighbor j.

3.1 Some Implementation Issues

We have tacitly assumed that the "adjustment period" ($Y \approx C - T$) is long enough so that node i has time to transmit for T units of time (according to the local clock), to receive all of the transmissions of its neighbors, and to calculate the value of expression (3). In practice, the time required to calculate the latter is negligible relative to other factors, and thus not a significant concern. And because in theory no upper bound need be placed on the value of Y (it merely acts as a "default" idle period length that may stretch based on relative clock drift rates), the need to have received transmissions from neighbors is also not a practical constraint, provided local clocks have a reasonable degree of accuracy.

The requirement that a node has time to transmit for T units of time, however, could be problematic. A cycle start time, as determined by (3), may require that a node begin transmitting for a particular cycle before that node has finished transmitting for the previous cycle. Increasing

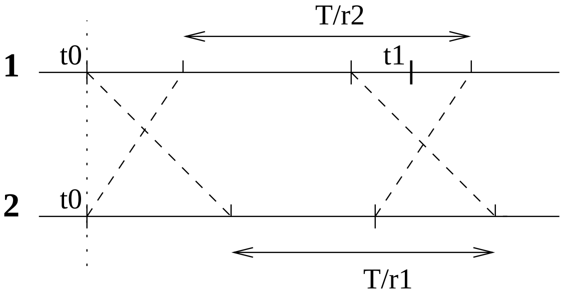

Figure 1: Time line demonstrating the effects of link latencies on computed start time. Latencies have the effect of stretching the cycle time C, making it highly unlikely that a computed start time is less than a node's perceived value of T.

the default value of Y eliminates this problem, but wastes bandwidth, since nodes do not transmit during the idle period. We resolve this by modifying the algorithm so that the default idle period length is zero, and have nodes set their start time to the value

$$\begin{aligned} &\max\left\{ T, \frac{1}{|U_i|}\sum_j \left(s_j^i(k) + \frac{f_j^i(k) - s_j^i(k)}{T}C \right) \right\} \\ &= \max\left\{ T, \frac{1}{|U_i|}\sum_j f_j^i(k) \right\}. \qquad (4) \end{aligned}$$

(The equality results from setting the default value of C equal to T, a result of setting the default idle period length to zero.) Expression (4), when translated to global time, becomes

$$\max\left\{ \frac{T}{r_i}, \frac{1}{|U_i|}\sum_j \left(s_j(k) + \ell_{ij} + \frac{C}{r_j} \right) \right\} \qquad (5)$$

This form of the algorithm clearly eliminates the cost of bandwidth wasted by a nonzero idle period. Notably, simulation results for the original algorithm (with idle period length set to zero) and modified algorithms are indistinguishable. The reason is that both algorithms reach steady state quickly (within 400 cycles for every simulation run in every configuration tested), and in the steady state, the value determined by (5) is rarely equal to the T/r_i maximand. For all simulated systems in which any link latency was greater than 100 nanoseconds, the maximum value in steady state was *never* the T/r_i term.

Figure 1 offers an explanation. We assume for simplicity that both nodes start a cycle at global time t_0, and consider the computation performed by node 1. The (global) time at which it should start the subsequent cycle is the average of $t_0 + T/r_1$ and $t_0 + T/r_2 + \ell_{12}$. The maximum in (5) will equal T/r_1 only if

$$\frac{1}{2}\left(2t_0 + \frac{T}{r_1} + \frac{T}{r_2} + \ell_{12} \right) \leq t_0 + \frac{T}{r_1},$$

or equivalently, if

$$\ell_{12} \leq \frac{T}{r_1} - \frac{T}{r_2}. \qquad (6)$$

If the r_i are assumed to satisfy $|1-r_i| < \delta_1$, for δ_1 a fixed positive constant, then $T/r_1 - T/r_2 < 2T\delta_1/(1-\delta_1^2)$, so (6) can never be satisfied if $\ell_{12} > 2T\delta_1/(1-\delta_1^2)$. For a transmit time of $T = 1$ ms. and $\delta_1 = 5 \times 10^{-5}$, this limits ℓ_{12} to at most 100 ns. Even in those rare cycles when the values of the algorithms differ, the difference falls well within the range of "noise" error.

Another subtlety that arises when the default length of the idle period is set to zero concerns the timing conflicts that can arise between the time at which information is needed, and the time at which it arrives. Figure (1) helps illustrate the potential problem. In the diagram, it appears that node 1 should start the transmission for the subsequent cycle at time t_1. At t_1, however, node 1 lacks information on the finish time of the transmit period on node 2, information required to compute the value of t_1. To avoid this, a node computing the start time of cycle $k+1$ uses the neighbor transmit times observed during cycle $k-1$ when performing the averaging. Node start time for cycle 1 is simply the start time for cycle 0 plus C. With this adjustment, expression (3) becomes

$$\frac{1}{|U_i|}\sum_j \left(s_j^i(k) + \frac{f_j^i(k-1) - s_j^i(k-1)}{T} C \right). \tag{7}$$

Since this revision is only necessary when $T = C$, we can simplify (7) to

$$s_i^i(k+1) = \frac{1}{|U_i|}\sum_j (s_j^i(k) + f_j^i(k-1) - s_j^i(k-1)) \tag{8}$$

Converting this to global time yields, for $k = 1, 2, \ldots$,

$$\begin{aligned} s_i(k+1) &= \\ &= \frac{1}{|U_i|}\sum_j \{(s_j(k) + \ell_{ij}) + \\ &\quad +[(s_j(k-1) + \frac{C}{r_j} + \ell_{ij}) - (s_j(k-1) + \ell_{ij})]r_i\} \\ &= \frac{1}{|U_i|}\sum_j \left(s_j(k) + \ell_{ij} + \frac{r_i}{r_j}C \right). \end{aligned} \tag{9}$$

(with $s_i(1) = s_i(0) + T/r_i$). Thus a global time representation of a realistic implementation of the algorithm, which necessarily involves a "maximum" function as explained above, is given by

$$\begin{aligned} s_i(k+1) &= \\ &= \max\left\{ \frac{T}{r_i}, \frac{1}{|U_i|}\sum_j \left(s_j(k) + \ell_{ij} + \frac{r_i}{r_j}C \right) \right\} \end{aligned} \tag{10}$$

for $k = 1, 2, \ldots$, with $s_i(1) = s_i(0) + T/r_i$.

The simulation results presented in Section 5 are based on the algorithm given by (10), because this emulates the timing interactions and constraints that would be observed in a practical implementation. The analysis in Section 4, however, focuses on the algorithm (3). Analysis of (10) is tractable, but messy because of the *max* expression. Analysis of (9) is similar to analysis of (3), but different initial conditions create additional transient terms in the former, and both produce identical steady state results. Including these technicalities would only obscure the analysis, while adding little to the understanding of the behavior of the algorithm. It is important to remember that despite these minor adjustments, fundamentally all that is occurring here are nodes using locally observable information to estimate the next-cycle start times of themselves and their neighbors, and then setting their next-cycle start time to the average of these values.

4 Analysis

At a moment of synchronization, node i knows the starting and ending times of the latest observed data transmission period of each of its neighbors. Based on these values, node i estimates the next cycle start times, and and adjusts its own start time to match the average of these estimates. This synchronization may result in cycle lengths for a particular node varying between cycles. To account for this, we introduce the term $c_i(k)$, which denotes the length, measured in global time of the kth cycle on node i.

In this section we develop and solve a system of difference equations that determine cycle start times as measured in global time. These are then used to derive a formula for individual cycle lengths.

We begin by writing equation (2) in a linear form. Toward that end, we introduce the adjacency matrix H that captures the network topology:

$$H_{ij} = \begin{cases} 1, & j \in U_i \\ 0, & \text{otherwise} \end{cases}$$

Using H, equation (2) can be rewritten as

$$s_i(k+1) = \frac{1}{\sum_j H_{ij}} \sum_j H_{ij}\left(s_j(k) + \frac{C}{r_j} + \ell_{ij} \right). \tag{11}$$

For convenience of computation, we define the stochastic adjacency matrix G by

$$G_{ij} = \frac{H_{ij}}{\sum_j H_{ij}} = \frac{H_{ij}}{\text{degree of node } i} \tag{12}$$

where the degree of node i includes the count of the self-loop. Using G, (11) becomes

$$s_i(k+1) = \sum_{j=1}^{N} G_{ij}\, s_j(k) + \sum_{j=1}^{N} G_{ij}\left(\frac{C}{r_j} + \ell_{ij} \right), \tag{13}$$

where N denotes the number of nodes in the network. Now, the second term in equation (13) is time-invariant. Let $\hat{B}$ be the matrix whose entries are given by $\hat{B}_{ij} = \frac{C}{r_i} + \ell_{ji}$, and let B be the $N \times 1$ column matrix given by $B = \text{diag}(G\hat{B})$, where the "diag" operator creates an $N \times 1$ matrix from an $N \times N$ matrix by extracting the diagonal elements of the larger matrix. That is, for an $N \times N$ matrix $M = [m_{ij}]$, diag(M) is the column vector with entries $m_{11}, m_{22}, m_{33}, \ldots, m_{NN}$. If we let $s(k)$ denote the $N \times 1$ column matrix whose entries are the $s_i(k)$, then (13) can be rewritten in matrix form:

$$s(k+1) = G\ s(k) + B. \tag{14}$$

A closed form solution for this equation is

$$s(k) = G^k s(0) + \sum_{i=0}^{k-1} G^i B, \quad k = 0, 1, 2, \ldots \tag{15}$$

where $s(0)$ represents the vector of global start times for the nodes. Equation (15) thus describes cycle start times as functions of the initial conditions $s(0)$ and network parameters.

Now, the length of cycle k at node i, measured in global time, is

$$c_i(k) = s_i(k+1) - s_i(k), \tag{16}$$

which, if we let $c(k)$ denote a column vector in a manner analogous to $s(k)$, can be written in matrix form as

$$c(k) = s(k+1) - s(k).$$

Substituting (14) in (16), and letting I denote the $N \times N$ identity matrix, we have

$$c(k) = (G - I)s(k) + B. \tag{17}$$

Finally, substituting (15) in (17) gives, after expansion and elimination of common terms,

$$c(k) = \left(G^{k+1} - G^k\right) s(0) + G^k B. \tag{18}$$

Thus we see that the properties of $c(k)$ depend significantly on the properties of the powers of G.

We show in the following section that under the present assumptions about network topology, the powers of G converge to a stochastic matrix Q, all of whose rows are identical. Thus, as $k \to \infty$, $c(k)$ approaches QB, a column vector in which all entries are the same. So, in the first order model, the cycle lengths at each node, as measured in global time, converge to the same value.

Moreover, if the r_j are near 1, and the l_{ij} are near zero, then each element of $\hat{B}$ is approximately equal to C. Now, G is a row normalized adjacency matrix derived from the adjacency matrix corresponding to a strongly connected network topology, from which we can conclude both that each row of G must contain a nonzero entry, and that each nonzero entry in a given row is equal to any other nonzero entry in that same row. As a consequence, left multiplication of $\hat{B}$ by G results in a matrix in which each entry is the average of some subset of elements of $\hat{B}$. Since each element of $\hat{B}$ is approximately C, it must be that each element of B is approximately C as well. If we let $\hat{C}$ denote the column vector in which each element is identically C, then as $k \to \infty$,

$$c(k) \to Q\ B \approx Q\ \hat{C} = \hat{C}. \tag{19}$$

That is, the value to which the cycle lengths converge is approximately C. We will determine the exact value of the limit in the next section. At this point, however, it is clear that the extent to which the limiting cycle length can vary from C is determined (in the present model) by the clock rates of individual nodes, the link latencies, and the topology of the network.

4.1 Convergence Behavior of G^k

Given that the behavior of the cycle lengths depends on the convergence properties of G^k, we need to determine the conditions under which G^k converges, the limit when it does converge (and the sense in which we mean "limit"), and the rate of convergence.

We begin with the issue of the conditions under which G^n converges. G is a finite stochastic matrix, and thus it must be the transition matrix for a finite Markov chain. Specifically, let $\mathcal{G}$ be the graph that represents the topology of our Cyclone network (including self-loops), and consider the Markov process corresponding to a "traveler" moving randomly (along edges of G) from node to node on $\mathcal{G}$, with the system in state S_i at a particular time epoch if the traveler is located at node i during that epoch. G is the transition matrix for this finite Markov chain. Because $\mathcal{G}$ is connected and contains self-loops, the Markov chain is ergodic (irreducibility follows from being connected, aperiodicity from the self-loops). Among other properties, ergodicity guarantees [KMST58] that the powers G^k approach a matrix Q, in the sense that each entry of G^k approaches the corresponding entry of Q. Moreover, each row of Q is the same positive probability vector W, where W is the unique probability vector such that $WG = W$. Equivalently, if $W = [w_1\ w_2\ \ldots\ w_N]$, then the w_i are uniquely determined through the system of equations

$$\sum_{i=1}^{N} w_i = 1; \qquad w_j = \sum_{i=1}^{N} w_i G_{ij}, \quad j = 1, 2, \ldots, N.$$

Since w_i represents the "long term" probability that the system is in state i, it is fairly intuitive that each w_i should be given by

$$w_i = \frac{\text{degree of node i}}{\text{number of nonzero entries of } G}$$

where here the degree of a node includes a count of the self-loop. A straightforward calculation verifies this result. To simplify notation, we will refer to the number of nonzero entries of G as the *degree* of G, denoted degree(G), and denote the degree of node i by degree(i), so that

$$w_i = \frac{\text{degree}(i)}{\text{degree}(G)}.$$

Given this value of the limiting matrix Q, let us consider equation (19) in more detail. Recall that $\hat{B}$ was defined by $\hat{B}_{ij} = \frac{C}{r_i} + \ell_{ji}$. Thus $B_i = (G\hat{B})_{ii}$ is given by

$$B_i = (G\hat{B})_{ii} = \sum_{m=1}^{N} G_{im}\hat{B}_{mi} = \sum_{m=1}^{N} \frac{H_{im}}{\text{degree}(i)} \left(\frac{C}{r_m} + \ell_{im}\right).$$

The limiting cycle length value, $W \cdot B$ with W as above, is thus

$$
\begin{aligned}
W \cdot B &= \sum_{i=1}^{N} w_i B_i = \\
&= \sum_{i=1}^{N} \frac{\text{degree}(i)}{\text{degree}(G)} \sum_{m=1}^{N} \frac{H_{im}}{\text{degree}(i)} \left(\frac{C}{r_m} + \ell_{im} \right) \\
&= \frac{1}{\text{degree}(G)} \sum_{i=1}^{N} \sum_{m=1}^{N} H_{im} \left(\frac{C}{r_m} + \ell_{im} \right) \\
&= \frac{1}{\text{degree}(G)} \sum_{i=1}^{N} \sum_{m=1}^{N} H_{im} \left(\frac{C}{r_m} \right) \\
&+ \frac{1}{\text{degree}(G)} \sum_{i=1}^{N} \sum_{m=1}^{N} H_{im} \ell_{im}
\end{aligned} \tag{20}
$$

From section 2.2, the expression $\frac{C}{r_i}$ represents the value, as measured in global time, of an interval that node i believes has length C. The first summand in the last expression is a weighted average of such terms, with increasing weight given to nodes with higher degree. The second summand in this last expression is easily seen to be the mean link latency in the network.

Determining the rate of convergence is relatively straightforward in theory: the powers of the transition matrix of an ergodic Markov chain converge at a rate related to the moduli of the eigenvalues of the matrix. This can be observed by considering the *spectral representation* of G. Specifically, the (positive integral) powers of a diagonalizable stochastic matrix G are given by

$$G^k = Q + \lambda_1^k A_1 + \lambda_2^k A_2 + \cdots + \lambda_m^k A_m \tag{21}$$

where the λ_i are the non-one eigenvalues of G, Q is the limit of the powers of G, and the A_i are differential matrices (i.e. each row of the matrix sums to zero) satisfying the following:

1) $A_i A_j = A_j A_i = 0$ if $i \neq j$
2) $A_i^k = A_i$, $1 \leq i \leq m$, $k = 1, 2, \ldots$.
3) $A_i Q = Q A_i = 0$, $1 \leq i \leq N$.
4) $\|A_i\| \leq 1$, $i = 1, 2, \ldots, m$.

Viewed in this form, it is clear that G^k converges at the same rate as the largest of the moduli of the λ_i. We refer to an eigenvalue with largest moduli as a "submaximal" eigenvalue. That is, an eigenvalue is submaximal if its absolute value is equal to the maximum of the moduli of the set of non-one eigenvalues (note that 1 is an eigenvalue of any stochastic matrix, and that Q would be the corresponding matrix in the spectral representation). Since G^k converges, it is clear that the moduli of the λ_i must be less than one. This also follows from one form of the Perron-Frobenius theorem [IM76] which also asserts that the eigenvalue one has multiplicity one.

The matrix G corresponding to the networks under consideration here is diagonalizable. To see this, consider the non-normalized adjacency matrix H corresponding to G. Because all links in the underlying network are bidirectional, H is symmetric, and thus diagonalizable (and all of its eigenvalues are real). Normalizing (as defined by (12)) amounts to multiplying H on the left by a diagonal matrix D with strictly positive diagonal entries. Such a D must be invertible and have an invertible square root. Thus $G = DH$ is similar to $D^{-\frac{1}{2}}(DH)D^{\frac{1}{2}} = D^{\frac{1}{2}}HD^{\frac{1}{2}}$. Since this last matrix is symmetric, G is similar to a symmetric matrix and thus it is diagonalizable and has only real eigenvalues.

Determining the eigenvalues of a matrix can be difficult. There are a few classes of topologies for which an explicit closed form solution for the eigenvalues can be found. One of these is a complete graph, in which the eigenvalues are easily seen to be 1 with multiplicity one (of course) and 0 with multiplicity $N-1$. The star topology is another whose eigenvalues are relatively easy to determine. We call a graph an *N-star* if the graph contains a total of N nodes: one "hub" node and $N-1$ leaf nodes. It can be shown that the non-one eigenvalues of the N-star are $\frac{1}{2}$ with multiplicity $N-2$ and $\frac{1}{N} - \frac{1}{2}$ with multiplicity one. Since $|\frac{1}{N} - \frac{1}{2}| < \frac{1}{2}$ for $N > 1$, the submaximal eigenvalue is $\frac{1}{2}$, and G^k converges at the rate of $\mathcal{O}\left(\frac{1}{2^k}\right)$.

Although we are more concerned with upper bounds on convergence rates of G^k, it is of interest to observe a situation in which a lower bound on the convergence rate can be computed. Specifically, if each node in $\mathcal{G}$ has degree m, then each nonzero entry of G is $\frac{1}{m+1}$, and trace$(G) = \frac{N}{m+1}$. Now, the trace of a matrix is equal to the sum of its eigenvalues, so since 1 is an eigenvalue of multiplicity one, the sum of the remaining eigenvalues is $\frac{N}{m+1} - 1 = \frac{N-m-1}{m+1}$. Let λ be a submaximal eigenvalue of G. In order for the sum of the non-one eigenvalues to equal the expression above, we must have

$$(N-1)|\lambda| \geq \frac{N-m-1}{m+1},$$

or equivalently

$$|\lambda| \geq \frac{N-m-1}{(m+1)(N-1)}. \tag{22}$$

Thus, the *fastest* that G^k can converge in this case is at the rate of

$$\left(\frac{N-m-1}{(m+1)(N-1)} \right)^k.$$

Finally, if we allow graphs with unidirectional links, then in terms of convergence, a worst case is a "one way" ring. Graphs with unidirectional links can have complex eigenvalues. The one way ring with N nodes has submaximal eigenvalue $\frac{1}{2} exp\left(\frac{2\pi i}{N}\right) + \frac{1}{2}$, with modulus $\frac{1}{2}\sqrt{2\cos\left(\frac{2\pi}{N}\right) + 2}$. By choosing N large enough, one can make this modulus arbitrarily close to 1.

Although this final example shows that convergence rates in theory can be relatively slow, this is not a practical limitation. As our simulation results show (for a bidirectional ring—a situation for which computing closed form expressions for eigenvalues can be daunting), convergence rates will generally be slower than more favorable topologies, but still well within tolerable limits.

4.2 Phase Difference Analysis

Up to this point, our analysis has focused on cycle lengths. Of equal importance, however, is the issue of phase drift. Specifically, how large can the gap become between the time a node begins a cycle and the time that the node begins receiving a transmission from a neighbor in that same cycle. While Cyclone networks can tolerate significant phase drift, this should never reach the point where nodes are so far out of phase that they are effectively operating on different cycles. We show here that with appropriate constraints on local clock drift rates and link latencies, phase drift remains bounded within tolerable limits. Since our analysis requires assumptions on the permissible clock drift and link latency rates, we assume for the remainder of this article that all clock rates r_i and all link latencies ℓ_{ij} satisfy

$$|1 - r_i| \le \delta_1 \text{ and } 0 \le \ell_{ij} \le \delta_2. \tag{23}$$

We discuss practical values for these bounds in later sections. As a notational convenience, let $\mathrm{var}(A)$, for a column vector A with elements $a_i, a_2, \ldots, a_N$, denote the quantity $\max_{i,j}\{|a_i - a_j|\}$.

The phase difference $p(k)$ at cycle k can be expressed in our model as

$$p(k) = \max_{i,j}\{|s(k)_i - (s(k)_j + \ell_{ij})|\}. \tag{24}$$

Since

$$\begin{aligned} p(k) &\le \max_{i,j}\{|s(k)_i - s(k)_j|\} + \max_{i,j}\{\ell_{ij}\} \\ &\le \max_{i,j}\{|s(k)_i - s(k)_j|\} + \delta_2 \\ &= \mathrm{var}(s(k)) + \delta_2, \end{aligned} \tag{25}$$

we restrict our analysis to $\mathrm{var}(s(k))$. It is instructive to see how this quantity behaves when G corresponds to a network graph that is complete. In this case, the value of each element of G is $\frac{1}{N}$, so left multiplication of a column vector by G results in a column vector whose entries are identical and equal to the average of the entries of the original column vector. Thus, any summand of $s(k)$ that has G as a left factor will contribute nothing to the value of $\mathrm{var}(s(k))$. Rewriting (15) as

$$s(k) = G^k s(0) + B + \sum_{i=1}^{k-1} G^i B, \tag{26}$$

we see that the only possible phase shift contributor is the matrix B, and that the expression we need to consider is

$$\mathrm{var}(B) \equiv \max_{i,j}\{|B_i - B_j|\}. \tag{27}$$

Now, recall that $B \equiv \mathrm{diag}(G\hat{B})$, where $\hat{B}_{ij} = \frac{C}{r_i} + \ell_{ji}$. By (23), the elements of $\hat{B}$ satisfy

$$\frac{C}{1+\delta_1} \le \hat{B}_{ij} \le \frac{C}{1-\delta_1} + \delta_2. \tag{28}$$

Because left multiplication by G is essentially performing weighted averaging (regardless of whether it corresponds to a complete graph) and the diag operator performs no scaling, (28) implies that

$$\mathrm{var}(B) \le \frac{2\delta_1}{1-{\delta_1}^2} C + \delta_2. \tag{29}$$

For the more complicated cases in which G does not correspond to a complete graph network topology, the intuition is similar. As k grows large, the powers of G approach a matrix Q whose rows are identical. Left multiplication of a column matrix by a matrix with this property yields a column matrix in which all entries are identical, and thus one which contributes nothing to phase drift. The tail end terms of the sum on the right side of equation (15) reach a point where they are contributing little, since successive terms are essentially multiplied on the left by Q. Thus, after a point (that depends, among other things, on the network topology), the potential additional phase drift added with each cycle becomes negligible.

The details of the general case analysis are presented in Appendix A, where it is shown that the phase difference satisfies

$$\begin{aligned} p(k) &\le 2\epsilon \sum_{j=1}^{m} |\lambda_j|^k \\ &+ \left(1 + 2\sqrt{N}\sum_{j=1}^{m} \frac{|\lambda_j|}{1-|\lambda_j|}\right)\left(\frac{2\delta_1}{1-{\delta_1}^2} C + \delta_2\right). \end{aligned} \tag{30}$$

Like the cycle length expression (20), this bound depends on the topology of the network via the eigenvalues of G. In particular, smaller eigenvalues give a tighter bound on phase drift, and quicker decay of the transient term. Also, as expected, greater clock rate accuracy, as reflected in smaller values of δ_1 results in a smaller phase drift bound.

4.3 Including Error Effects

The assumptions implicit in our analysis have been more generous than can be expected in a real system using primarily off the shelf components. Specifically, we assumed that node clock clock rates and link latencies remained constant, and we ignored effects of discretization errors due to limited clock granularities. These assumptions are now relaxed, with the introduction of several random variables corresponding to potential timing discrepancies occurring during each cycle. These random variables do not correspond to individual sources of error, but instead correspond to sources and recipients of data. Specifically, for each k (corresponding to cycle k), and i, j satisfying $1 \le i, j \le N$, let $e^i_j(k)$ be a random variable with zero mean and "small" variance, and such that $e^i_j(k)$ represents the variation due to error effects of the information that node i receives about node j during cycle k. In addition, let $E(k)$ be the $N \times N$ matrix with entries $E_{ij} = e^j_i$. Incorporating these error terms in our original node operation expression (2), yields

$$s_i(k+1) = \frac{1}{|U_i|} \sum_j \left(s_j(k) + \ell_{ij} + \frac{C}{r_j} + e_j^i(k) \right). \quad (31)$$

That is, the nodes perform the same algorithm as earlier, but the information used in each cycle can be perturbed by the value of the error terms. Since the system corresponding to (2) converges (and shows no signs of bordering on chaotic behavior) one should expect that if the error terms are relatively small, then the augmented system should possess "convergence like" behavior. This is indeed the case. Intuitively, the algorithm "corrects" for errors in estimation of the start times since the estimated start time of any cycle is based only on the data transmission start and finish times received during the previous cycle. In this way, errors that occur during a given cycle do not accumulate.

The details of the analysis of (31) are presented in Appendix B, where it is shown that as $k \to \infty$,

$$c(k) = QB + A,$$

where Q is the limit of G^k, and $A = [a_{ij}]$ is a matrix satisfying $|a_{ij}| < \left(1 + \sqrt{2N} \sum_{j=1}^m |\lambda_j - 1| \frac{1}{1-|\lambda_j|}\right) \epsilon$, for $1 \leq i, j \leq N$. That is, cycle lengths become bounded within a region surrounding the ideal model limit. The "radius" of this region is bounded, with the bound depending on the magnitudes of the eigenvalues and the bound on the magnitude of the error terms. Appendix B also presents the error model analogue of the phase difference bound (31):

$$\begin{aligned} p(k) \leq\ & 2\epsilon \sum_{j=1}^m |\lambda_j|^k + 2\delta_2 + \\ & + \left(1 + 2\sqrt{N} \sum_{j=1}^m \frac{|\lambda_j|}{1-|\lambda_j|}\right) \left(\frac{2\delta_1}{1-{\delta_1}^2} C + \delta_2\right) + \\ & +2\left(1 + \sqrt{N} \sum_{j=1}^m \frac{|\lambda_j|}{1-|\lambda_j|}\right) \epsilon. \end{aligned} \quad (32)$$

4.4 Interpretation

The theoretical bounds (40) and (32) are valuable because they prove that the cycle lengths and phase differences converge toward steady state values and then remain close to those values. In addition, they provide clues to the degree to which various parameters affect these limiting values. The expressions are much less useful as predictors of the variations that can be expected in a working implementation. For example, if the network topology is a chain of rings with 20 nodes (see Figure 2), and we assume values of $C = 10^{-3}$, $\delta_1 = 5 \times 10^{-5}$, $\delta_2 = 2 \times 10^{-5}$, and $\epsilon = 10^{-7}$, then the estimate of the phase bound given by (32) is approximately 7.6ms, or almost eight times as long as the cycle time C. In simulation, however, every run with these parameters and this configuration had steady state phase differences less than 42μs.

There are several reasons for this. First, our analysis applies to a relatively large class of matrices. In many instances, knowledge of the exact topology and spectral decomposition of the corresponding row-normalized adjacency matrix would allow for a much sharper estimate of parameter variability. Even if the specific configuration and system parameters are known, our analysis provides a worst case, rather than average case, bound. Since our analysis consists of a number of steps, each of which employs a number of such estimates, it should not be surprising that the numbers seen in simulation can be orders of magnitude smaller than the estimated bounds.

5 Simulation Results

Our theoretical analysis has shown that in the absense of errors in the transmission of timing information, cycle lengths stabilize at a fixed and predetermined value. Introduction of timing errors results in cycle lengths fluctuating near this same value. Based on these results, we expect that observed cycle lengths and phase differences will be affected by the following parameters:

- Timing errors, including variations in clock drift rates, variations in link latencies, discretization errors, and other timing errors. These effects are captured in the theoretical model by the parameter ϵ.
- Link latencies, whose magnitudes are bounded by δ_2 in the model.
- Network topology, which consists of both network configuration and the number of nodes. Topological effects are captured in the model by the values of the eigenvalues of the row-normalized adjacency matrix G.
- Clock rates, which are effectively constrained by δ_1 in the model.

This sections describes the results of a series of tests on a simulated Cyclone system with nodes performing the algorithm described in global terms by (10). Our goal is to both demonstrate the feasibility of the synchronization algorithm and determine the relationships between various input parameters and performance of the algorithm.

5.1 Setup

The values we study in the experiments are the cycle length jitter and the maximum phase difference. The jitter measures the degree of consistency between consecutive cycles on a node, while the phase difference measures the degree of synchronization between cycles on different nodes. Specifically, if node i starts cycle k at time $s_i(k)$, then the length of cycle k on node i is $c_i(k) = s_i(k+1) - s_i(k)$. The jitter on node i at cycle k is defined as $j_i(k) = |c_i(k) - c_i(k-1)|$, and the "maximum" jitter on the network at cycle k is defined as $\max_i j_i(k)$. The system maximum phase difference is measured in stages. The maximum phase difference for node i during cycle k is given by $\max_{j \in U_i} |s_i(k) - (s_j(k) +$

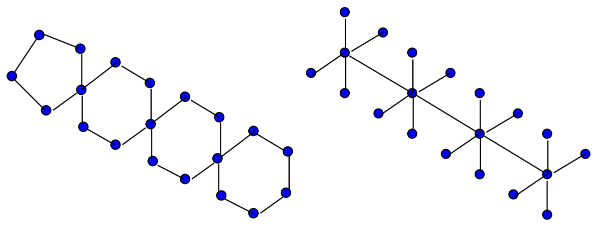

Figure 2: Ring chain (left) and stars configurations with 20 nodes.

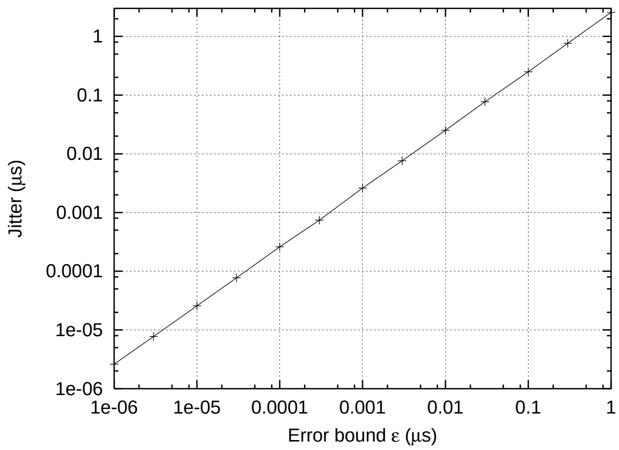

Figure 3: Jitter: Stable region. Values of jitter depend on the error range. The observed stable region value of jitter grows linearly with the size of error range for all networks, all clock rate drifts, and various link latencies. In this figure, results are shown for the 30-node ring chain with a fixed set of latencies ranging from 0 to 10 μs ($\delta_2 = 10$) and clock rate drifts of up to 50 ppm ($\delta_1 = 5 \times 10^{-5}$). The value at each data point is the maximum of 10 experiments.

$\ell_{ij})|$. Taking the maximum for all nodes provides the maximum phase difference for cycle k, and taking the maximum of these over all cycles provides the system maximum phase difference for the simulation run.

Each simulation uses a transmission period length of $T =$ 1 ms. We set the default idle period to zero, as explained in Section 3.1, and thus the default value of C is also 1 ms. We have run several simulations for more than one million cycles (1000 seconds), but present results here from runs of only up to 10,000 cycles, since all experiments show that system phase difference and cycle length jitter reach their region of stability in less than 200 and 400 iterations respectively.[1] All reported time parameters are values as measured in global time.

To observe the effect of random errors, we slightly perturb the values of arrival times as recorded by destination nodes. These perturbations encompass several sources of error, including conversion of the real-valued time to the integral "ticks" by each clock (discretization error), variations in local clock rates, and changing link latencies. For simulation purposes, we consider all perturbations to be independent random errors uniformly distributed in the interval $(-\epsilon, \epsilon)$.

To capture the variation in latencies from link to link, we assigned point-to-point latencies to the links in each configuration. The range of latencies varied in different sets of experiments. The values of latencies were chosen randomly from among the permitted range, with the exception that we forced at least one link latency to assume the maximum permissible value in each individual trial.

Every simulation run assumed one of five network topologies: *clique* (a complete graph), *ring*, *chain* (a string of nodes in which the endpoints are not connected to each other), *ring chain* (a chain of rings connected with a bridging node) and *stars* (a chain of stars connected with a bridging link). The two latter configurations are illustrated on Figure 2. Mathematical analysis suggests that in terms of convergence rates, the best and worst configurations are the clique and chain, respectively. In practice, LANs are often constructed in the form of rings or stars, and the whole network may look like a collection of interconnected LANs. This prompted our choice of the ring chain and stars topologies. For each network configuration, we varied the number of nodes from 10 to 40, in increments of 10 nodes. The individual stars in the collection of connected stars and the individual rings in the collection of connected rings consisted of 5 or 6 nodes each, with the number of bridging nodes between 1 and 8.

Each clock in the system has a steady rate, which however may be slightly different from the 1 second per second rate of the perfect clock. Modern clocks that can be used on nodes may be assumed to be accurate within 50 parts per million ([Sym]). This corresponds to a clock drift rate parameter (δ_2) value of 5×10^{-5}. We have also repeated experiments with other values of δ_1.

5.2 Summary of Results

The maximum observed value of cycle length jitter in the stable region grows with the timing error ϵ as $\alpha\epsilon$, where α lies between 1 and 3; the effects of the other parameters are negligible. This should not be surprising in light of the results from Section 4.1, which show that in the absence of timing errors, cycle length jitter approaches zero exponentially. System maximum phase difference, on the other hand, is significantly dependent on both network topology and link latency values, but is unaffected by either timing errors or clock drift rates, within realistic ranges for these parameters. In the following subsections, we examine the influence of these parameters individually.

5.3 Timing Errors

Effect on jitter. As expected, the values of transmission errors have a strong effect on the jitter exhibited by the network. In fact, the maximum (as well as average) value of the jitter grows linearly with the range of errors (see Figure 5). The speed of convergence is not affected, but a system with larger error magnitudes reaches its region of stability

[1] Note the difference between the terms "stable region" and "steady-state region". For smaller ϵ values, cycle lengths are constrained to a smaller steady-state "envelope" surrounding the limiting value, and it typically takes more iterations to fall within this smaller region. For practical purposes, however, once cycle lengths remain within 100ns of the limiting value, they can be considered to be in the stable region, even though they may not yet have reached the steady state.

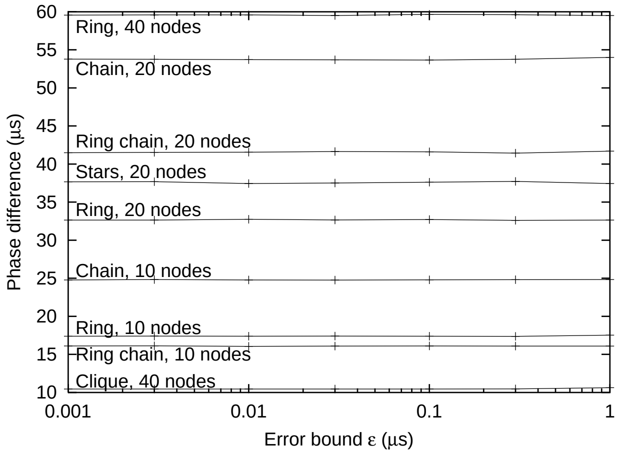

Figure 4: System maximum phase difference is independent of the error bound ϵ. Graphs are shown for several topologies, each with a fixed set of link latencies ranging from 0 to 10 μs ($\delta_2 = 10$) and clock rate drifts of up to 50 ppm ($\delta_1 = 5 \times 10^{-5}$). The value at each data point is the maximum of 10 experiments.

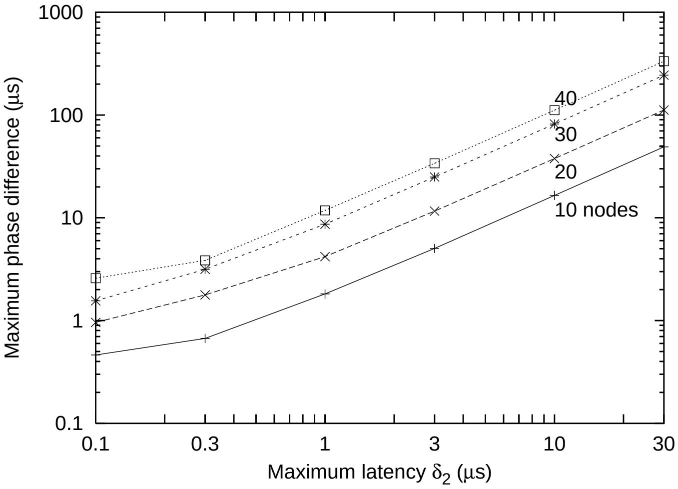

Figure 6: Maximum phase difference grows roughly linearly with increasing maximum link-latency δ_2. Graphs are shown for stars configurations. Each data point is the maximum of 200 trials, 20 trials using each of ten different sets of link latencies with ranges from 0 to δ_2. Error bound ϵ is fixed at 0.1 μs and clock drift rates are up to 50 ppm ($\delta_1 = 5 \times 10^{-5}$).

Size	Configuration	$b \times 10^{-3}$	$a \times 10^{-2}$
40	Stars	10.4	2.4
	Ring chain	10.8	2.9
	Ring	8.24	2.08
	Chain	2.1	0.46
30	Stars	18.4	10
	Ring chain	70.2	320
	Ring	14.8	4.74
	Chain	3.83	1.0
20	Stars	40.3	28
	Ring chain	680	135
	Ring	330	45
	Chain	8.8	1.8
10	Stars	129	1720
	Ring chain	112	137
	Ring	136	2010
	Chain	38.1	21.7

Table 1: Exponential descend in jitter evolution: $y = ae^{-bx}$. Parameters a, b are estimated from experiments with latencies ranging from 0 to 10μs ($\delta_2 = 10$) and clock rate drifts up to 50 ppm ($\delta_1 = 5 \times 10^{-5}$).

quicker, because larger errors imply a larger convergence "envelope" about the limiting cycle length value. These effects are observed for values of timing errors up to 10μs.

The system converges at an exponential rate, which is consistent with the theoretical prediction (see section 4.1). The rate of convergence depends on the network topology. Figure 5 gives an illustration of this behavior. Visualization of the convergence allows us to estimate parameters of the exponential descend. Selected results are shown in table 1.

Effect on phase difference. Timing errors have no discernible effect on the values of the maximum phase difference. In testing sample phase differences for several configurations and numbers of nodes, phase difference remains virtually unchanged as values of the timing error parameter ϵ are varied. In our trials, we used link latencies in the range of 0 to 10 μs ($\delta_2 = 10$ μs) and a clock drift rate bound of $\delta_1 = 5 \times 10^{-5}$. Trials with different values for latencies and clock drift rates showed the same pattern. The results were derived for each specific configuration, network size, and error parameter value, by choosing ten different sets of latencies in the permissible range, running twenty trials with each latency set, and then choosing the maximum phase difference observed during the total of 200 trials. We repeated experiments with values of timing parameters lying in the range from 0 to 1 μs. In practice, ϵ can be expected to be less than 100 ns.

5.4 Link Latencies

Effect on jitter. Due to a delay ℓ_{ij} on the link from node j to node i, timing information takes ℓ_{ij} time to reach i from j. The feedback from node j will eventually have node i start its cycles closer to node j, mitigating the discrepancy. Intuitively, system settling time should grow with increasing differences between the delays on the "forward" paths leading from i and those on the "backward" paths leading to i. To test this, we ran experiments with several sets of latencies, with varying maximum and minimum values and maximum to minimum ratios. In all cases, the system settled at the same rate. Moreover, latency ranges show no discernible effect on cycle length jitter.

Effect on phase difference. Our experiments show, however, that latency values, and specifically the maximum allowed link latency, have a large influence on the observed system phase difference. As Figure 6 illustrates, our results indicate a roughly linear relationship, which is consistent with the theoretical prediction (see fermula 32 in

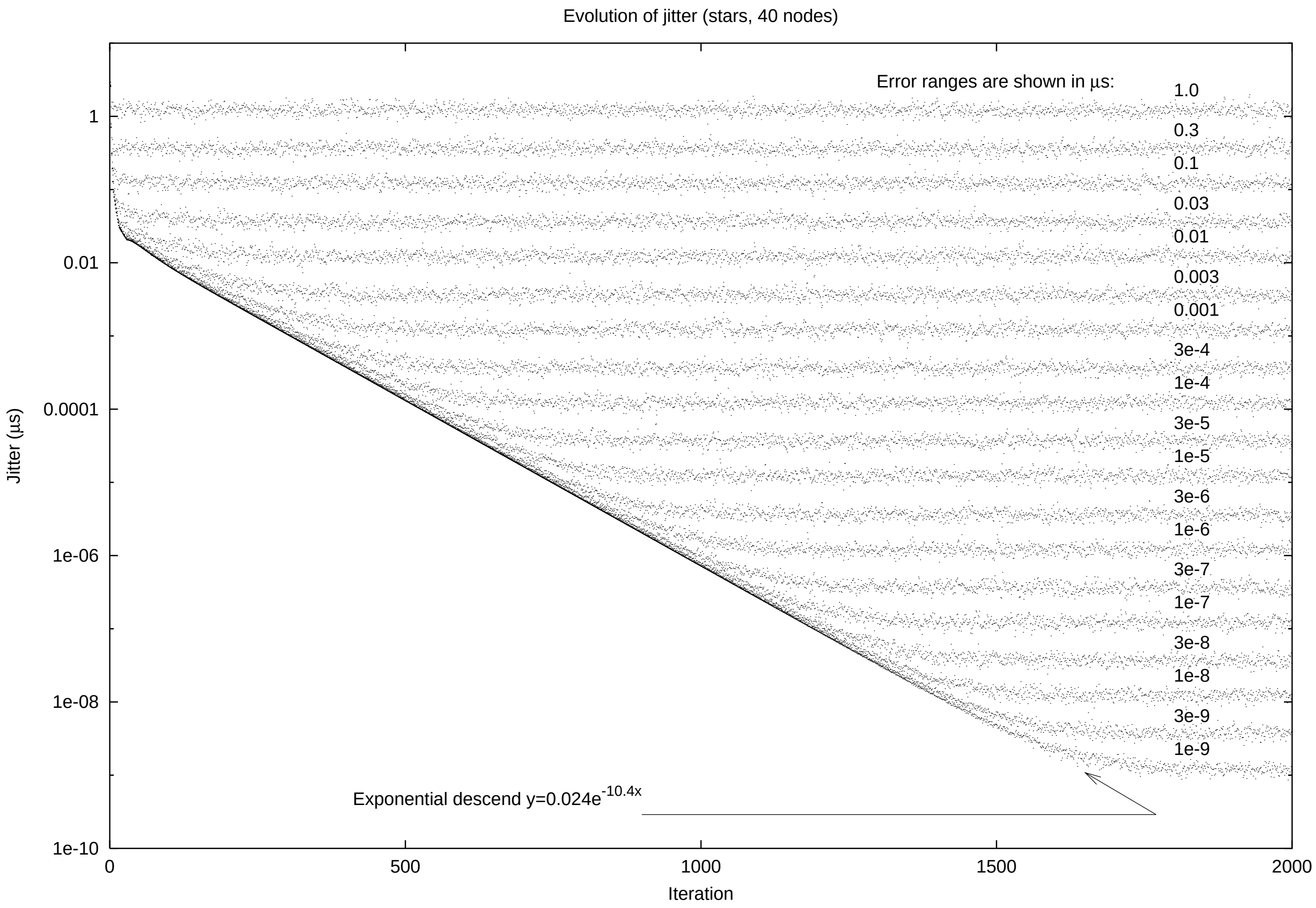

Figure 5: Jitter: Exponential descend toward steady state. The system reaches stability at an exponential rate. In this figure, graphs are shown for the 40-node stars configuration with a fixed set of latencies ranging from 0 to 10 μs ($\delta_2 = 10$) and clock rate drifts of up to 50 ppm ($\delta_1 = 5 \times 10^{-5}$).

section 4.3). Trials with configurations other than stars show similar patterns, with the exception of the completely connected graph, for which phase differences appear to be independent of both latencies and network configuration. This is not surprising in light of the comments above, since in the clique, the set of "forward" and "backward" latencies are identical at each node in the network.

5.5 Network Topology

Effect on jitter. In the steady state region, the ring, chain, stars, and ring chain topologies exhibit a similar amount of jitter, while the clique consistently performs at a lower jitter (see Figure 7). The small graph diameter of the clique, along with high average node degrees indicate that both factors play a role in the jitter experienced in the system. Since the remaining four configurations experience similar jitter, despite increasing network diameter, it appears that diameter is not a major factor, and that the superior performance of the clique can be attributed to its significant advantage in average node degree. In addition, the jitter experienced by the clique decreased as the number of nodes increased, despite an unchanged network diameter. These observations are corroborated by the theoretical results from Section 4, which showed that in the absence of timing errors, the steady state jitter for all of the configura-

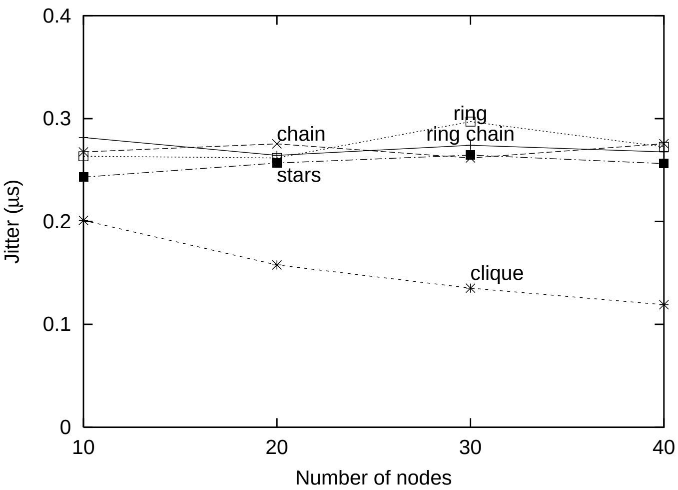

Figure 7: Jitter is loosely dependent on node degree, but appears to be independent of graph diameter. Graphs are shown for several topologies, each with a fixed set of link latencies in the range of 0 to 10 μs ($\delta_2 = 10$), error bound $\epsilon = 0.1$ μs, and clock drift rates of up to 50 ppm ($\delta_1 = 5 \times 10^{-5}$). The value at each data point is the maximum over 100 trials.

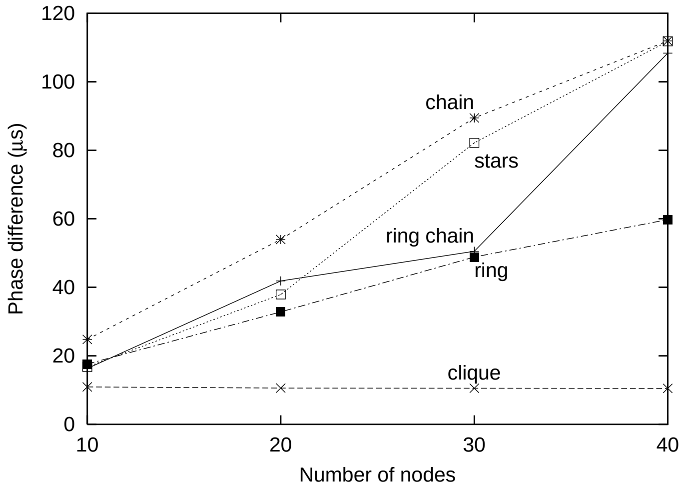

Figure 8: Maximum phase difference depends on both the system configuration and number of nodes. Graphs are shown for several topologies, each with a fixed set of link latencies in the range of 0 to 10 μs ($\delta_2 = 10$), error bound $\epsilon = 0.1$ μs, and clock drift rates of up to 50 ppm ($\delta_1 = 5 \times 10^{-5}$). The value at each data point is the maximum over 100 trials.

tions is zero. In a clique configuration, the introduction of a timing error that may cause a relatively large difference in cycle lengths between neighboring nodes is immediately communicated to all other nodes. The averaging of the information from all nodes has the effect of smoothing the deviation due to the timing error, and also "damping" the phase difference. In configurations with fewer communication paths, the averaging is performed over a much smaller number of nodes, leaving the potential for greater maximum phase differences.

As mentioned above (section 5.3), the rate of convergence of the system to a steady state is exponential and depends on the network topology, in accordance with the theoretical prediction in section 4.1. The clique, whose underlying network forms a complete graph, converges in just a few iterations. All other studied networks converged at an exponential rate $y = ae^{-bx}$, where x is the iteration count and y is the value of the system jitter. Parameters a and b, estimated from the experiments, are collected in table 1.

Effect on phase difference. With regard to phase difference, both network configuration and network size play a significant role. As Figure 8 illustrates, the clique topology performs best, while the chain, also as expected, performs worst. The ring showed phase differences at levels that were approximately the mean of the clique and chain values. Ring chain and stars perform, for the most part, somewhere between the ring and chain topologies. For all but the clique topology, size was a significant influence, with phase differences increasing roughly linearly with the number of nodes.

Intuitively, topologies with greater connectivity should perform better, since node timing information is quickly disseminated. Indeed, the clique, with the best possible connectivity, clearly outperforms the others. But our experiments show that the relationship between topology and algorithm performance is much more subtle, and cannot be explained by simple quantities such as network diameter and average node degrees. As a rough guide, however, our experiments clearly demonstrated that smaller diameter and greater connectivity result in improved algorithm performance, as measured in terms of both jitter and phase difference.

5.6 Clock Rates

A primary goal of the cycle synchronization algorithm is to reduce timing discrepancies between nodes. Variations in clock rates are one source of these discrepancies. While modern time measuring equipment can be extremely accurate, terminal nodes are expected to have inexpensive, and therefore less accurate, clocks. The ITU requirement for customer premises equipment (CPE) clocks in SONET, for example, is for accuracy no worse than 5×10^{-5} ([Sym]). We performed experiments in which minimum clock accuracy requirements for nodes varied from 0 to 5×10^{-4} (with all other factors unchanged). Neither cycle length jitter nor phase difference were substantially affected by the changes in the values of clock rates. This is consistent with our theoretical results, which predict that (1) under the assumption that clock rates remain constant, jitter converges to zero exponentially (assuming clock rates are "reasonable"), and that (2) even in the presence of timing errors, phase difference is only slightly influenced by the clock drift rate parameter δ_1. We thus conclude that the cycle synchronization algorithm removes any effect that varying clock rates might have on timing discrepancies between the nodes.

6 Conclusions

We have presented a cycle synchronization algorithm for Cyclone networks. Fundamentally, the algorithm allows nodes to use locally observable information to estimate the next-cycle start times of themselves and their neighbors, and then sets their next-cycle start time to the average of these values. A practical system implementation requires some minor adjustments to the "pure" algorithm, such as the inclusion of a maximum function that guarantees that nodes do not begin a cycle before completing the transmission phase of the prior cycle, and the use of information from earlier cycles to guarantee that information required for processing is available at the time of computation. Because none of these potential modifications affect steady state system performance, we analyzed a pure version of the algorithm for clarity, and ran simulated experiments using a practical implementation version of the algorithm

Our theoretical analysis shows that under ideal conditions, the algorithm results in system cycle lengths that converge to a fixed and known (if link latencies and clock rates are known and constant) value, and that this convergence occurs at an exponential rate that depends on the network configuration. The effect of network configuration is captured in our analysis by the magnitudes of the eigenvalues of the row-normalized adjacency matrix corresponding to the graph describing the network topology. In the presence of timing errors, variations in local clock rates and link latencies, and other minor perturbations, cycle lengths converge to an interval centered on the ideal-model cycle length limit, with the radius of this interval depending on several parameters including the allowed variations in clock drift rates, allowed variation in link latencies, and the magnitude of the noise error bound. The rate of convergence in this case, as in the ideal model, is exponential and dependent on network configuration. It is important to note that the the theoretical bounds obtained in our analysis are not sharp, since the wide variety of potential network topologies that need to be explained leads to a relatively large class of adjacency matrices under consideration.

Simulation results on a practical implementation version of the algorithm show that under all LAN configurations tested, the system reaches steady state quickly (in under 400 iterations), and that both cycle length jitter and system phase differences remain well within tolerable limits. The maximum observed steady state cycle length jitter for practical parameter values was on the order of 100ns, and the observed phase difference was never more than 110 μs even for the impractical *chain* configurations simulated for their “worst-case” behavior. In general, the cycle length jitter depends primarily on the allowable timing error magnitude ϵ, growing in the stable region as $\alpha\epsilon$, where $1 < \alpha < 3$; other factors, such as network topology, link-latency ranges, and clock drift rates, have little effect on the observed jitter, within the parameter ranges tested. Phase differences, on the other hand, are significantly dependent on both network topology and link latency values, but are unaffected by either timing errors or clock drift rates, within realistic ranges for these parameters. From a practical point of view, this indicates that the algorithm is well suited for local area networks, where relatively short links, small latencies, and relatively large timing errors (as compared to long haul fiber networks) are the norm.

A Appendix: Phase Differences Under Ideal Conditions

Our subsequent analysis is helped by viewing $\mathbb{C}^N$ as an N-dimensional Hilbert space with Euclidean norm (e.g. if $\overline{x}$ = is a row or column vector with elements $x_1, x_2, \ldots, x_N$, then $\|\overline{x}\|^2 = \sum_{i=1}^N |x_i|^2$), and $N \times N$ matrices as elements of $\mathcal{B}(\mathbb{C}^N)$, the space of bounded linear operators on $\mathbb{C}^N$ (see [Con85] for a good introduction to these concepts). Using these conventions, if $A = [a_{ij}]$ is an $N \times N$ matrix and $\overline{x}$ is as above, then any entry of $A\overline{x}$ has the form $\sum_{j=1}^N a_{ij}x_j$. But,

$$|\sum_{j=1}^N a_{ij}x_j| \le \sum_{j=1}^N |a_{ij}||x_j| \le \|A\| \sum_{j=1}^N |x_j|,$$

so if the moduli of the x_i are bounded by some constant Γ, the moduli of the elements of $A\overline{x}$ are bounded by $\|A\|N\Gamma$, and consequently

$$\mathrm{var}(A\overline{x}) \le 2\|A\|N\Gamma. \qquad (33)$$

Now, substituting the spectral representation (21) for G into equation (15) and observing that the Q term contributes nothing to phase distortion, it follows that two expressions must be addressed:

$$(\lambda_1^k A_1 + \lambda_2^k A_2 + \cdots + \lambda_m^k A_m)s(0) \qquad (34)$$

and

$$\sum_{i=1}^{k-1}(\lambda_1^i A_1 + \lambda_2^i A_2 + \cdots + \lambda_m^i A_m)B. \qquad (35)$$

Assume that the absolute values of the elements of $s(0)$ are bounded by ϵ. Then by (33), expression (34) contributes no more than

$$2(|\lambda_1|^k + |\lambda_2|^k + \cdots + |\lambda_m|^k)\epsilon = 2\epsilon\sum_{j=1}^m |\lambda_j|^k$$

to the phase difference. Since the λ_i are all less than one in modulus, this expression has only a transient effect on phase drift.

The analysis of the contribution from (35) is more involved. We first show that for an $N \times N$ differential matrix $A = [a_{ij}]$, and an $N \times 1$ matrix $E = [e_i]$ with positive entries,

$$\mathrm{var}(AE) \le 2\sqrt{N}\|A\|\mathrm{var}(E). \qquad (36)$$

(Recall that a differential matrix has the property that the sum of the elements of any row is zero. Note also that the above inequality is significantly stronger than (33).) The proof is based on the observation that if all of the elements of E are identical, then the differential property of A means that AE is the zero column vector, and more important for the present purpose, that $\mathrm{var}(AE) = 0$. Intuitively then, if the elements of E are not identical, but $\mathrm{var}(E)$ is small, then $\mathrm{var}(AE)$ should be small as well. So, fix i and consider $(AE)_i = \sum_{l=1}^N a_{il}e_l$. Define the sets $\mathcal{P}$ and $\mathcal{N}$ by

$$\mathcal{P} = \{l : a_{il} > 0\} \qquad \mathcal{N} = \{l : a_{il} < 0\}.$$

Clearly $(AE)_i = \sum_{l\in\mathcal{P}} a_{il}e_l + \sum_{l\in\mathcal{N}} a_{il}e_l$. Since the e_l are positive,

$$\begin{aligned}&\min_l\{e_l\}\sum_{l\in\mathcal{P}} a_{il} + \max_l\{e_l\}\sum_{l\in\mathcal{N}} a_{il}\\ &\le \sum_{l\in\mathcal{P}} a_{il}e_l + \sum_{l\in\mathcal{N}} a_{il}e_l\\ &\le \max_l\{e_l\}\sum_{l\in\mathcal{P}} a_{il} + \min_l\{e_l\}\sum_{l\in\mathcal{N}} a_{il}.\end{aligned}$$

That is, $(AE)_i$ is restricted to an interval of length

$$\begin{aligned}&(\max_l\{e_l\} - \min_l\{e_l\})\sum_{l\in\mathcal{P}} a_{il}\\ &+(\min_l\{e_l\} - \max_l\{e_l\})\sum_{l\in\mathcal{N}} a_{il}\end{aligned}$$

$$\begin{aligned} &= (\max_l\{e_l\} - \min_l\{e_l\}) \left(\sum_{l \in \mathcal{P}} a_{il} - \sum_{l \in \mathcal{N}} a_{il} \right) \\ &= 2\mathrm{var}(E) \sum_{l \in \mathcal{P}} a_{il}. \end{aligned}$$

the last equality resulting from the differential property of A. Now, let f_i be the vector with ith entry one, and all other entries zero. Since the norm of a real matrix and its transpose are equal,

$$\sum_{l \in \mathcal{P}} {a_{il}}^2 \le \sum_{l=1}^{N} |a_{il}|^2 = \|A^t f_i\|^2 \le \|A^t\|^2 \|f_i\|^2 = \|A\|^2.$$

It is a straightforward exercise in constrained optimization to show that this implies that $\sum_{l \in \mathcal{P}} a_{il}$ is bounded by $\sqrt{N}\|A\|$. Thus, $(AE)_i$ is restricted to an interval of length $2\sqrt{N}\|A\|\mathrm{var}(E)$, from which (36) follows.

We now apply this to equation (35). Since clearly $\mathrm{var}(V+W) \le \mathrm{var}(V) + \mathrm{var}(W)$ for arbitrary column vectors V and W, and since the A_i and B satisfy the prerequisites for relation (36), it follows that

$$\begin{aligned} &\mathrm{var}\left(\sum_{i=1}^{k-1} (\lambda_1^i A_1 + \lambda_2^i A_2 + \cdots + \lambda_m^i A_m) B \right) \le \\ &\le \sum_{i=1}^{k-1} \mathrm{var}((\lambda_1^i A_1 + \lambda_2^i A_2 + \cdots + \lambda_m^i A_m) B)) \\ &\le \sum_{i=1}^{k-1} |\lambda_1|^i \mathrm{var}(A_1 B) + |\lambda_2|^i \mathrm{var}(A_2 B) + \cdots \\ &\qquad \cdots + |\lambda_m|^i \mathrm{var}(A_m B) \\ &\le \sum_{i=1}^{k-1} (|\lambda_1|^i + |\lambda_2|^i + \cdots + |\lambda_m|^i) 2\sqrt{N} \mathrm{var}(B) \\ &< \sum_{i=1}^{\infty} (|\lambda_1|^i + |\lambda_2|^i + \cdots + |\lambda_m|^i) 2\sqrt{N} \mathrm{var}(B) \\ &< 2\sqrt{N} \left(\frac{2\delta_1}{1 - {\delta_1}^2} C + \delta_2 \right) \sum_{j=1}^{m} \frac{|\lambda_j|}{1 - |\lambda_j|}, \end{aligned}$$

with the last inequality resulting from (29).

Combining the computed bounds for (27), (34), and (35), along with relation (25) proves (31).

B Appendix: Analysis Assuming Timing Errors

We analyze equation (31) in a manner similar to the ideal case analysis. In the present situation, the analogue of equation (14) is:

$$s(k+1) = Gs(k) + B + \mathrm{diag}(GE(k)), \tag{37}$$

with closed form solution

$$s(k) = G^k s(0) + \sum_{i=0}^{k-1} G^i B + \sum_{i=0}^{k-1} G^i \mathrm{diag}(GE(k-i)). \tag{38}$$

The corresponding cycle length equation is

$$\begin{aligned} &c(k) = s(k+1) - s(k) = \\ &= G^{k+1} s(0) + \sum_{i=0}^{k} G^i b + \sum_{i=0}^{k} G^i \mathrm{diag}(GE(k+1-i)) \\ &- \left(G^k s(0) + \sum_{i=0}^{k-1} G^i B + \sum_{i=0}^{k-1} G^i \mathrm{diag}(GE(k-i)) \right) \\ &= (G^{k+1} - G^k) s(0) + G^k B + \left(\sum_{i=0}^{k} G^i \mathrm{diag}(GE(k-i)) \right. \\ &\qquad \left. - \sum_{i=0}^{k-1} G^i \mathrm{diag}(GE(k-i-1)) \right). \end{aligned}$$

The first two terms on the right side of this last equation are well understood from the ideal case analysis. The last term, however, is new to the augmented model. By rearranging terms, it can be rewritten as

$$\mathrm{diag}(G\, E(k)) + \sum_{i=0}^{k-1} (G^{i+1} - G^i)\ \mathrm{diag}(GE(k-i-1)). \tag{39}$$

Using the spectral representation for G,

$$\begin{aligned} &G^{i+1} - G^i = \\ &= Q + \lambda_1^{i+1} A_1 + \lambda_2^{i+1} A_2 + \cdots + \lambda_m^{i+1} A_m \\ &\quad - \left(Q + \lambda_1^i A_1 + \lambda_2^i A_2 + \cdots + \lambda_m^i A_m \right) \\ &= (\lambda_1^{i+1} - \lambda_1^i) A_1 + (\lambda_2^{i+1} - \lambda_2^i) A_2 + \cdots \\ &\quad \cdots + (\lambda_m^{i+1} - \lambda_m^i) A_m \end{aligned}$$

so that

$$\begin{aligned} &\sum_{i=0}^{k-1} (G^{i+1} - G^i)\ \mathrm{diag}(GE(k-i-1)) = \\ &= \sum_{i=0}^{k-1} \big((\lambda_1^{i+1} - \lambda_1^i) A_1 + (\lambda_2^{i+1} - \lambda_2^i) A_2 + \cdots \\ &\qquad \cdots + (\lambda_m^{i+1} - \lambda_m^i) A_m \big)\ \mathrm{diag}(GE(k-i-1)) \\ &= \sum_{j=1}^{m} \sum_{i=0}^{k-1} (\lambda_j^{i+1} - \lambda_j^i) A_j \mathrm{diag}(GE(k-i-1)). \end{aligned}$$

Taking the norm of this last expression yields

$$\begin{aligned} &\| \sum_{j=1}^{m} \sum_{i=0}^{k-1} (\lambda_j^{i+1} - \lambda_j^i) A_j \mathrm{diag}(GE(k-i-1)) \| \le \\ &\le \sum_{j=1}^{m} \| \sum_{i=0}^{k-1} (\lambda_j^{i+1} - \lambda_j^i) A_j \mathrm{diag}(GE(k-i-1)) \| \\ &\le \sum_{j=1}^{m} \sum_{i=1}^{k} |\lambda_j^{i+1} - \lambda_j^i| \|A_j\| \|\mathrm{diag}(GE(k-i-1))\|. \end{aligned}$$

Now, assume that the $e_j^i(k)$ are uniformly bounded by ϵ. (Earlier we used ϵ as a bound for the magnitude of the elements of $s(0)$. This can be considered the same value, since the elements of $s(0)$ are essentially the "errors" that occur because the nodes are not perfectly synchronized at system start time.) Then,

$$\|\mathrm{diag}(GE(k-i-1)\|^2 = \sum_{l=1}^{N}(GE(k-i-1))_{ll}^2 \leq 2N\epsilon^2,$$

so that $\|\mathrm{diag}(GE(k-i-1)\| \leq \sqrt{2N}\epsilon$. Thus

$$\sum_{j=1}^{m}\sum_{i=0}^{k-1}|\lambda_j^{i+1}-\lambda_j^i|\|A_j\|\|\mathrm{diag}(GE(k-i))\| \leq$$

$$\leq \sum_{j=1}^{m}\sqrt{2N}\epsilon\sum_{i=0}^{k-1}|\lambda_j^{i+1}-\lambda_j^i| = \sum_{j=1}^{m}\sqrt{2N}\epsilon|\lambda_j-1|\sum_{i=0}^{k-1}|\lambda_j|^i$$

$$< \sum_{j=1}^{m}\sqrt{2N}\epsilon|\lambda_j-1|\frac{1}{1-|\lambda_j|},$$

where the convergence of the geometric series is guaranteed since the norms of the λ_i are all strictly less than one. Thus for all k

$$\begin{aligned} &\|\sum_{i=0}^{k-1}(G^{i+1}-G^i)\ \mathrm{diag}(GE(k-i-1))\| \\ &< \quad \sqrt{2N}\epsilon\ \sum_{j=1}^{m}|\lambda_j-1|\frac{1}{1-|\lambda_j|}. \end{aligned} \tag{40}$$

Combining the results for the augmented model, we see that

$$c(k) = (G^{k+1}-G^k)\ s(0) + G^k\ B + \mathrm{diag}(G\ E(k)) + \hat{A},$$

where $\hat{A}$ is a matrix whose entries (by (40)) are all less than a fixed multiple of ϵ in absolute value. Since the entries of $\mathrm{diag}(G\ E(k))$ are clearly bounded by ϵ, it follows that as $k \to \infty$,

$$c(k) = Q\ B + A,$$

where Q is the limit of G^k, and $A = [a_{ij}]$ is a matrix satisfying $|a_{ij}| < \left(1+\sqrt{2N}\sum_{j=1}^{m}|\lambda_j-1|\frac{1}{1-|\lambda_j|}\right)\epsilon$, for $1 \leq i,j \leq N$.

We turn now to the phase drift analysis of the augmented model. The point of departure is equation (38), and in particular, the term

$$\sum_{i=0}^{k-1}G^i\mathrm{diag}(GE(k-i)).$$

Since left multiplication by G amounts to averaging selected elements of the vector being multiplied, and since each of the elements of the $E(i)$ are bounded by ϵ, it follows that

$$\mathrm{var}(\mathrm{diag}(GE(i))) \leq 2\epsilon \qquad 1 \leq i \leq k.$$

Using the spectral representation of G and (36), yields

$$\begin{aligned} &\mathrm{var}\left(\sum_{i=0}^{k-1}G^i\mathrm{diag}(GE(k-i))\right) \leq \\ &\leq \quad \mathrm{var}(\mathrm{diag}(GE(k))) + \\ &\quad\quad + \ \mathrm{var}\left(\sum_{i=1}^{k-1}G^i\mathrm{diag}(GE(k-i))\right) \\ &\leq \quad 2\epsilon + \mathrm{var}\left(\sum_{i=1}^{k-1}(Q+\lambda_1^iA_1+\lambda_2^iA_2+\cdots\right. \\ &\quad\quad \left.\cdots+\lambda_m^iA_m)\mathrm{diag}(GE(k-i))\right) \\ &= \quad 2\epsilon + \mathrm{var}\left(\sum_{i=1}^{k-1}(\lambda_1^iA_1+\lambda_2^iA_2+\cdots\right. \\ &\quad\quad \left.\cdots+\lambda_m^iA_m)\mathrm{diag}(GE(k-i))\right) \\ &\leq \quad 2\epsilon + 2\sqrt{N}\epsilon\sum_{i=1}^{k-1}(|\lambda_1|^i+|\lambda_2|^i+\cdots+|\lambda_m|^i) \\ &< \quad 2\epsilon + 2\sqrt{N}\epsilon\sum_{j=1}^{m}\frac{|\lambda_j|}{1-|\lambda_j|} \\ &= \quad 2\left(1+\sqrt{N}\sum_{j=1}^{m}\frac{|\lambda_j|}{1-|\lambda_j|}\right)\epsilon. \end{aligned}$$

Combining this with the results from Section (4.2), proves (32).

Acknowledgement

The authors wish to thank G. W. Stewart for pointing out a concise proof of the diagonalizability of the row-normalized adjacency matrices considered in this paper.

References

[Con85] J.B. Conway, *A course in functional analysis*, Springer-Verlag, New York, 1985.

[Gor97] W. Goralski, *SONET-a guide to synchronous optical network*, McGraw-Hill, New York, 1997.

[How71] R.A. Howard, *Dynamic probabilistic systems: Markov models*, vol. 1, J. Wiley & Sons, 1971.

[IM76] D. Isaacson and R. Madsen, *Markov chains: Theory and applications*, J. Wiley & Sons, 1976.

[Kle75] L. Kleinrock, *Queueing systems*, vol. 1, J. Wiley & Sons, New York, 1975.

[KMST58] J.G. Kemeny, H. Mirkil, J.L. Snell, and G.L. Thompson, *Finite mathematical structures*, Prentice-Hall, 1958.

[Lam78] Leslie Lamport, *Time, clocks and ordering of events in a distributed system*, Communications of the ACM **21** (1978), no. 7, 558–565.

[Lee98] S. Y. Lee, *A study of Cyclone technology*, Ph.D. thesis, University of Maryland at College Park, 1998.

[Mil] D. Mills, *Bibliography on computer network time synchronization*, http://www.eecis.udel.edu/~mills/bib.htm.

[Mil92] D. L. Mills, *Network time protocol (version 3), specification, implementation, and analysis*, Tech. Report RFC-1305, University of Delaware, March 1992.

[Sym] *Synchronizing telecommunications networks: Basic concepts*, Tech. report, Symmetricom, http://www.symmetricom.com.