A Real-Time Implementation of Precise Timestamp-Free Network Synchronization

Max Li, Stefan Gvozdenovic, Alexander Ryan,
Radu David, and D. Richard Brown III
Dept. of Electrical and Computer Engineering
Worcester Polytechnic Institute
100 Institute Rd, Worcester, MA 01609
{mhli,sgvozdenovic,ajryan,radud,drb}@wpi.edu

Andrew G. Klein
Department of Engineering and Design
Western Washington University
516 High St, Bellingham, WA 98225
andy.klein@wwu.edu

***Abstract*—This paper describes a real-time implementation of precise synchronization between a master and slave node. The nodes were both implemented on separate Texas Instruments DSP boards with real-time software implemented in C. Timestamp-free synchronization is performed implicitly through the timing of the master node's responses to the slave node. The slave node tracks and predicts the master node's clock through precise time of arrival estimation and Kalman filtering of the estimates. Experimental results demonstrate the slave node is able to consistently predict the master node's clock offset to within a few nanoseconds over one second prediction intervals.**

***Index Terms*—synchronization, delay estimation, oscillator dynamics, real-time signal processing**

I. Introduction

A variety of synchronization protocols and systems have been developed over the last 30 years, including Network Time Protocol (NTP) [1], Precision Time Protocol (PTP) [2], the Global Positioning System (GPS) [3], and several "lightweight" synchronization protocols for sensor networks, e.g., [4]. A common theme running through almost all of these synchronization protocols and systems is that they are based on application-layer or MAC-layer exchanges of digital timestamps between nodes in the network. Digital timestamps inherently have a minimum resolution, limiting accuracy, and add overhead to the network traffic. Timestamp-free synchronization techniques were originally studied in the context of natural phenomena, e.g. synchronization of the firing rates of fireflies, and led to formal mathematical models for systems of pulse-coupled oscillators in [5], [6] and application of these models to wireless sensor networks in [7]–[9]. While these studies represented an exciting paradigm shift with respect to the prior work, a limitation of the pulse-coupled oscillator literature is that it is based on unidirectional transmissions and assumes negligible propagation delays. This inherently limits the synchronization accuracy of these methods.

This paper describes a real-time implementation and experimental results for the bidirectional timestamp-free synchronization protocol described in [10] and illustrated in Fig. 1. This protocol accounts for propagation delays and its accuracy is only limited by fundamental bounds of delay estimation. The real-time implementation described in this paper is based on the Texas Instruments TMS320C6713 DSP starter kit (DSK), audio-frequency signaling, and wired connections between the DSKs. Our results show that precise synchronization can be achieved between a pair of DSKs and serve as a proof of concept for potential implementation at radio frequencies and/or wireless channels.

This work was supported by the National Science Foundation awards CCF-1302104 and CCF-1319458 and by an equipment donation from Texas Instruments.

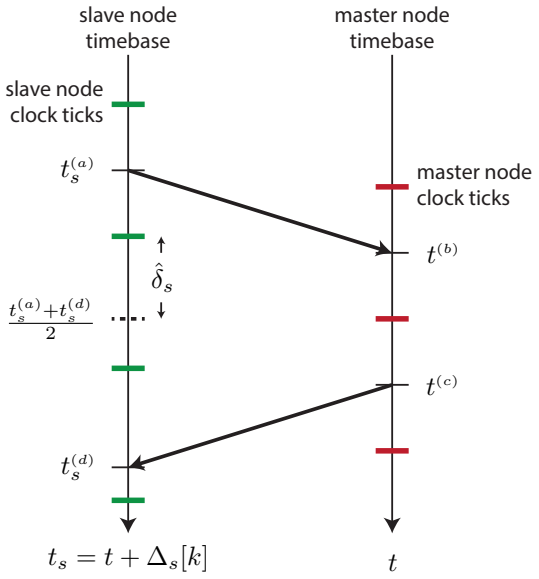

Fig. 1. Timestamp-free synchronization bidirectional signal exchange.

II. Timestamp-Free Synchronization Protocol

The timestamp-free synchronization protocol is described in detail in [10]. This section summarizes the key features of the protocol to provide context for the implementation methodology and experimental results in this paper.

Figure 1 shows the interactions between a slave node and the master node in exchange k. The time-varying clock offset at the slave node with respect to the master node is denoted as $\Delta_s[k]$ and local time the slave node is denoted as

$$t_s = t + \Delta_s[k]$$

where t is the reference time corresponding to the local clock at the master node. The timestamp-free synchronization protocol begins with the slave node transmitting a signal to the master node at arbitrary local time $t_s^{(a)}$. The signal arrives at the master node at local time

$$t^{(b)} = t_s^{(a)} - \Delta_s[k] + \tau_{S\to M}$$

where $\tau_{S\to M}$ is the propagation delay between the slave node and the master node. The master node then transmits a signal back to the slave node at time $t^{(c)}$ where $t^{(c)}$ is selected such that

$$\frac{t^{(b)} + t^{(c)}}{2} \pmod{T_0} = 0 \tag{1}$$

where T_0 is master node *clock tick period*. Note that, unlike the usual sender/receiver synchronization protocol, e.g. [11], no timestamps are exchanged between the nodes here. Implicit timing information is embedded in the master node's response to slave node by selecting $t^{(c)}$ so that a local clock tick of the master node is centered between $t^{(b)}$ and $t^{(c)}$. The slave node receives the reply signal from the master node at local time

$$t_s^{(d)} = t^{(c)} + \Delta_s[k] + \tau_{M\to S}.$$

The slave node can now estimate its clock tick offset with respect to the master node by calculating

$$\hat{\delta}_s = \left(\frac{t_s^{(a)} + t_s^{(d)}}{2}\right)_{T_0} \tag{2}$$

where the notation $(z)_{T_0}$ corresponds to wrapping z to the interval $[-T_0/2, T_0/2)$. In the absence of noise and if the propagation is reciprocal, i.e., $\tau_{S\to M} = \tau_{M\to S}$, the slave node can exactly estimate its clock offset with respect to the master node (modulo the clock period). To account for drift between the master and slave node, noisy clock offset estimates can also be provided as observations to a Kalman filter. The timestamp-free synchronization technique accounts for propagation delay, hence its accuracy is only limited by the fundamental bounds of delay estimation and the accuracy of the channel reciprocity assumption.

III. Delay Estimator for Bandpass Signals

A fundamental building block of the timestamp-free synchronization protocol is accurate delay estimation. This section provides an overview of the delay estimator used in our real-time implementation of the timestamp-free synchronization protocol.

We assume the signals exchanged between the nodes are bandpass signals and that frequency offsets are negligible. We further assume the synchronization pulse is a bandpass signal of the form

$$s(t) = \cos(\Omega_0 t)u(t)$$

where $u(t)$ is a bandlimited signal such that $U(\Omega) = 0$ for all $|\Omega| \geq \Omega_0$. The synchronization pulse in discrete time can be expressed as

$$\begin{aligned} s[\ell] &= \left[\cos(\Omega_0 t)u(t)\right]_{t=\ell T_s} \\ &= \cos(\omega_0 \ell)u[\ell] \end{aligned}$$

where $\omega_0 = \Omega_0 T_s$ is the normalized carrier frequency in radians/sample.

The discrete-time observation with unknown delay τ can be expressed as

$$\begin{aligned} y[\ell] &= \left[\cos(\Omega_0(t-\tau))u(t-\tau)\right]_{t=\ell T_s} \\ &= \cos(\omega_0(\ell - \tau f_s))u(\ell T_s - \tau) \end{aligned}$$

for $\ell = 0, \ldots, L-1$ where L is the number of samples in the observation. Standard cross correlation techniques with the template waveform $s[\ell]$ can be used to generate a quantized delay estimate $\hat{\ell} \in \mathbb{Z}$. The accuracy of this quantized delay estimate is limited, however, by the sampling rate of the delay estimator.

To refine the delay estimate, we define

$$\begin{aligned} s_i[\ell, \hat{\ell}] &= \cos(\omega_0(\ell - \hat{\ell}))u[\ell - \hat{\ell}] \\ s_q[\ell, \hat{\ell}] &= \sin(\omega_0(\ell - \hat{\ell}))u[\ell - \hat{\ell}] \end{aligned}$$

for $\ell = 0, \ldots, L-1$ where $\hat{\ell}$ is the quantized delay estimate obtained via discrete-time cross correlation. We can then compute

$$\begin{aligned} z_i[\hat{\ell}] &= \sum_{\ell=0}^{K-1} y[\ell] s_i[\ell, \hat{\ell}] \\ &= \frac{1}{2}\sum_{\ell=0}^{K-1} \Big(\cos(\omega_0(2\ell - \tau f_s - \hat{\ell})) + \cos(\omega_0(\tau f_s - \hat{\ell}))\Big) \\ &\qquad \times u(\ell T_s - \tau)u[\ell - \hat{\ell}] \\ &\approx \frac{1}{2}\cos(\omega_0(\tau f_s - \hat{\ell}))\sum_{\ell=0}^{K-1} u(\ell T_s - \tau)u[\ell - \hat{\ell}] \end{aligned}$$

where the approximation results from the fact that $\sum_{\ell=0}^{K-1} \cos(\omega_0(2\ell - \tau f_s - \hat{\ell}))u(\ell T_s - \tau)u[\ell - \hat{\ell}] \approx 0$. Similarly, we can calculate

$$\begin{aligned} z_q[\hat{\ell}] &= \sum_{\ell=0}^{K-1} y[\ell] s_q[\ell, \hat{\ell}] \\ &\approx \frac{1}{2}\sin(\omega_0(\tau f_s - \hat{\ell}))\sum_{\ell=0}^{K-1} u(\ell T_s - \tau)u[\ell - \hat{\ell}]. \end{aligned}$$

Note that the terms $z_i[\hat{\ell}]$ and $z_q[\hat{\ell}]$ represent in-phase and quadrature correlations at the quantized delay $\hat{\ell}$. We can then compute

$$\begin{aligned} \theta &= \tan^{-1}\left(\frac{z_q[\hat{\ell}]}{z_i[\hat{\ell}]}\right) \\ &\approx \tan^{-1}\left(\frac{\sin(\omega_0(\tau f_s - \hat{\ell}))}{\cos(\omega_0(\tau f_s - \hat{\ell}))}\right) \\ &= \omega_0(\tau f_s - \hat{\ell}). \end{aligned} \tag{3}$$

The refined delay estimate can then be computed as

$$\hat{\tau} = \left(\hat{\ell} + \frac{\theta}{\omega_0}\right) T_s$$

where θ is calculated according to (3).

IV. Real-Time Implementation

To demonstrate the feasibility of timestamp-free synchronization, the timestamp-free synchronization protocol was implemented with audio-frequency signals on Texas Instruments TMS320C6713 DSP starter kits (DSKs) with wired connections between the DSKs. One DSK was used for the master node and a second DSK was used for the slave node. The DSKs include an AIC23 stereo codec that provides ADC and DAC functionality with 16 bits per channel. A sampling rate of $f_s = 16000$ Hz was selected for the implementation.

A block diagram of the timestamp-free synchronization test setup is shown in Fig. 2. The right line in/out channels on the DSKs were used for exchanging synchronization pulses according to the timestamp-free synchronization protocol. The synchronization pulses used in all of the tests were modulated sinc pulses of the form

$$s(t) = \begin{cases} \cos(\Omega_0 t)\mathrm{sinc}(Wt) & \frac{-T}{2} \le t \le \frac{T}{2} \\ 0 & \text{otherwise} \end{cases}$$

with nominal carrier frequency $\frac{\Omega_0}{2\pi} = 4000$ Hz, bandwidth $W = 2\pi \cdot 200$ rad/sec, and duration $T = 70$ ms. The left line out channels were used for outputting clock pulses from the master and slave nodes for recording to a WAV file and synchronization performance evaluation as described in Section V-B. The clock pulses were also modulated sinc pulses with the same parameters as the synchronization pulses.

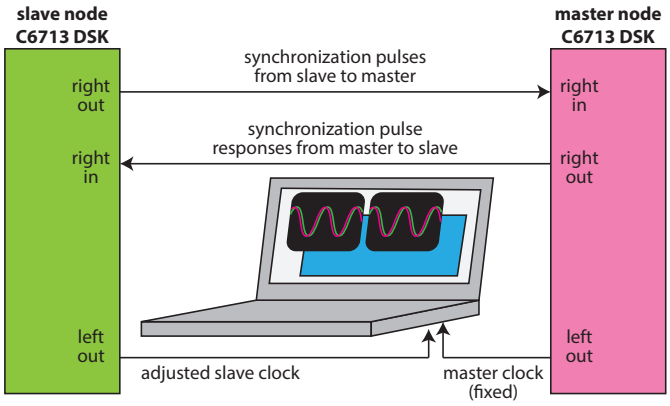

Fig. 2. Block diagram of the connections between the DSKs used for the master and slave nodes. Each DSK has a independent local oscillator.

The master node and slave node software was implemented in C. The local clock tick period at each node was nominally set to $T_0 = \frac{4096}{16000}$ sec. A state diagram of the master node is shown in Fig. 3. On the synchronization pulse channel, the master node uses non-coherent correlation in the searching state to detect the presence of a modulated waveform at Ω_0. If a signal is detected, the master node then records samples to a recording buffer with length slightly longer than the duration of the expected sync pulse. Once this buffer is full, the master node then counts the time until a local clock tick occurs (counting up) and subsequently counts down until the counter is again equal to zero. By outputting a time-reversed copy of the recording buffer, the master node effectively echoes back the sinc pulse from the slave node with synchronization pulses mirrored around the master node's local clock tick. Simultaneously, the master node also maintains a counter and outputs its local clock pulses on the clock pulse output channel.

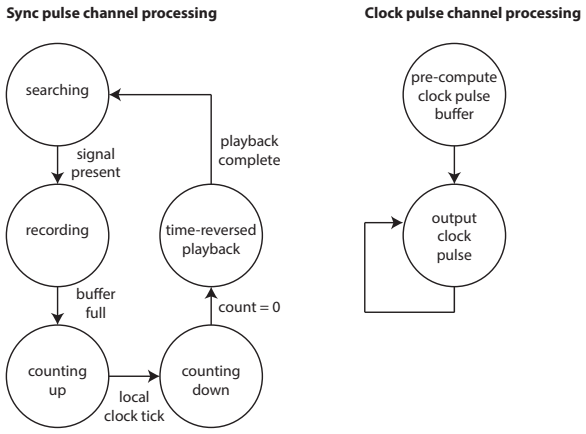

Fig. 3. State diagram for the master node.

A state diagram of the slave node is shown in Fig. 4. The slave begins by waiting for the appropriate time to transmit a synchronization pulse to the master. After the synchronization pulse has been transmitted, the slave then enters the searching and recording states (with the same functionality as described previously for the master node). When the recording buffer is full, the slave estimates the delay of the received signal according to the procedure in Section III and calculates the corresponding clock offset. The slave then asserts a flag to tell the Kalman filter a new observation is available. Simultaneously, the slave maintains predictions of the master node's clock. These predictions are informed by the clock offset observations from the timestamp-free synchronization protocol and account for drift between the master and slave. In each sampling period, the slave updates its predictions of the master clock, computes the clock pulse sample based on the predicted master clock time, and then outputs this synchronized clock pulse on the clock pulse output channel.

V. Experimental Results

This section summarizes our experimental results in a "noise-free" over-the-wire setting. Modulated sinc pulses were transmitted by the slave node to the master node with period $4T_0 \approx 1$ sec. A total of 24 tests were conducted with three different DSKs (labeled A, B, C) serving alternately as the master and slave nodes. Innovation statistics, drift estimates, and master/slave clock pulses were recorded in each test. The results of these tests are summarized below.

A. Innovations Statistics and Drift Estimates

The statistics of the innovations in a Kalman filter provide a standard method by which the performance of the Kalman filter can be evaluated in the absence of ground truth. The

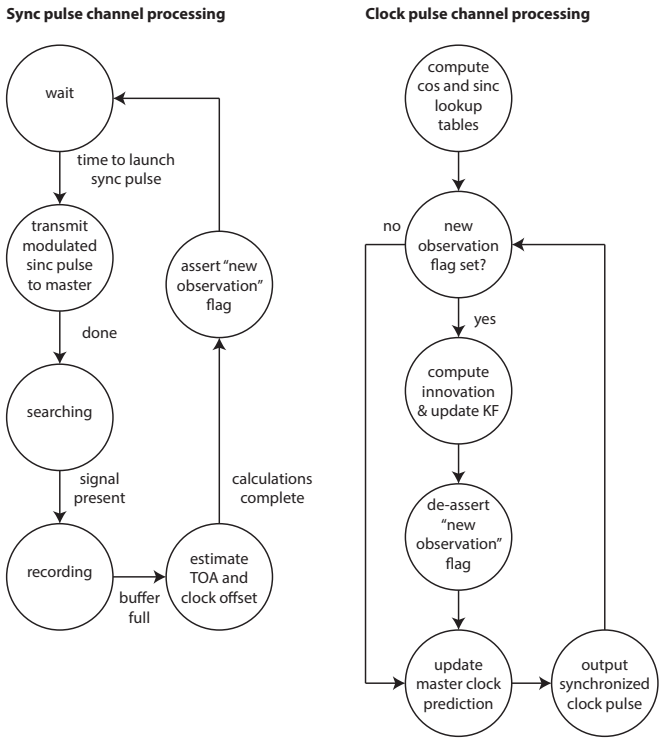

Fig. 4. State diagram for the slave node.

innovations are denoted as $\tilde{y}[k]$ and correspond to the difference between the predicted clock offset from the Kalman filter and the new clock offset observation (at the time of the new clock offset observation). Since, in our experiments, clock offset observations occur approximately once per second, the innovations represent the error in the slave node's predictions of the master node's clock approximately one second into the future.

To analyze the innovation statistics, each master/slave combination, i.e., A/B, B/A, A/C, C/A, B/C and C/B, was tested by running the timestamp-free synchronization protocol for approximately five minutes and collecting the final 200 innovations (corresponding to approximately the last 200 seconds of the test) by pausing the slave node and exporting the innovations save buffer to a file. Each master/slave combination was tested twice in this manner, resulting in a total of 12 innovation recordings. The mean and standard deviation of the saved innovations for each test are provided in Table I below. Averaged over all tests (a total of 2400 saved innovations), the mean innovation was approximately -0.54 ns and the mean standard deviation was approximately 2.1 ns. These results suggest that the slave node is able to accurately track and predict the master node's clock, at least over one second intervals, using the timestamp-free protocol.

Table I also provides the drift estimates at the slave node from the Kalman filter. The estimated drift was obtained by pausing the slave at the end of each test and noting the Kalman filter state estimates in the debugger. Note that the estimated drifts are consistent, e.g., the estimated drift of the A/C tests

TABLE I
SUMMARY OF INNOVATION STATISTICS AND DRIFT ESTIMATES.

M/S	Test Number	$\mathrm{mean}[\tilde{y}[k]]$	$\mathrm{std}[\tilde{y}[k]]$	Drift Estim.
A/B	1	-3.1 ns	1.2 ns	+6.96 ppm
A/B	2	0.3 ns	3.1 ns	+6.89 ppm
B/A	1	0.5 ns	2.6 ns	-6.97 ppm
B/A	2	1.2 ns	1.8 ns	-6.85 ppm
A/C	1	0.3 ns	1.7 ns	+18.52 ppm
A/C	2	-2.9 ns	2.0 ns	+18.69 ppm
C/A	1	0.8 ns	1.3 ns	-18.47 ppm
C/A	2	-0.3 ns	2.0 ns	-18.44 ppm
B/C	1	-0.5 ns	2.3 ns	+11.54 ppm
B/C	2	-0.0 ns	2.8 ns	+11.52 ppm
C/B	1	-1.6 ns	2.4 ns	-11.37 ppm
C/B	2	-0.9 ns	1.8 ns	-11.57 ppm

is approximately the sum of the estimated drifts in the A/B and B/C tests, and reciprocal, e.g., the estimated drift of the A/B tests is approximately opposite the estimated drift of the B/A tests.

B. Clock Pulse Synchronization Tests

In addition to collecting innovation statistics and drift estimates, the synchronized clock pulses transmitted on the left channels by the master and slave nodes were recorded to a WAV file for analysis of the offsets. To confirm that the recording equipment was sufficiently accurate for these tests, a recording equipment accuracy test was first performed by connecting both the left and right inputs of the recording computer to the left line output of a DSK running the master node program (without connecting the slave node). In this configuration, the master node simply outputs clock pulses with nominal period $T_0 \approx 0.25$ sec. After recording approximately 800 clock pulses, the left and right channels of the resulting WAV file were processed in Matlab with the maximum likelihood delay estimator described in Section III. The mean offset and standard deviation between the estimated delays of the pulses received on the left and right channels were calculated to be $\mathrm{mean}[\epsilon[k]] = 0.028$ ns and $\mathrm{std}[\epsilon[k]] = 0.32$ ns, respectively.

Figure 5 shows a histogram of the clock pulse offsets for the A/B:1 test. In this test, the mean estimated clock offset was $\mathrm{mean}[\epsilon[k]] = +36.3$ ns and the standard deviation of the estimated clock offsets was $\mathrm{std}[\epsilon[k]] = 1.3$ ns. Table II summarizes the results from the timestamp-free synchronization tests in all of the tested configurations. Note that the clock pulse offset standard deviations are similar to the standard deviations of the innovations in Table I.

Note that, unlike the innovation statistics, the clock pulse synchronization tests revealed repeatable mean offsets as large as 60 ns. These offsets are of the form

$$\mathrm{mean}[\epsilon[k]] \approx (X_{\mathrm{M/S}} - 11) \text{ ns}$$

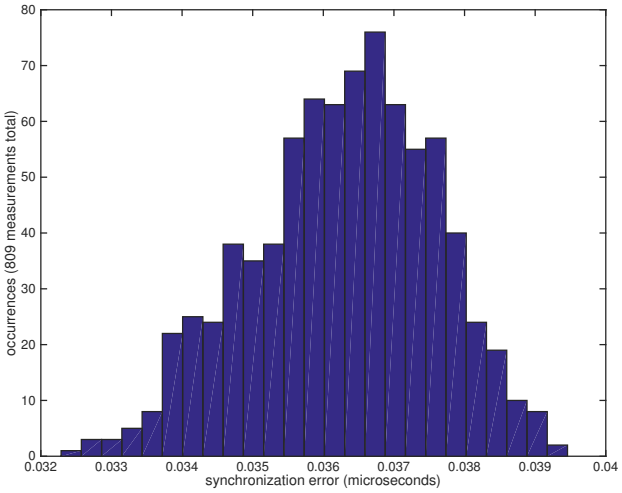

Fig. 5. Histogram of estimated clock pulse offsets for the A/B:1 test.

TABLE II
SUMMARY OF CLOCK PULSE SYNCHRONIZATION RESULTS.

M/S	Test Number	$\mathrm{mean}[\epsilon[k]]$	$\mathrm{std}[\epsilon[k]]$
A/B	1	+36.3 ns	1.3 ns
A/B	2	+38.8 ns	2.8 ns
B/A	1	-60.4 ns	2.1 ns
B/A	2	-59.6 ns	1.9 ns
A/C	1	+28.3 ns	1.8 ns
A/C	2	+25.3 ns	1.7 ns
C/A	1	-48.4 ns	1.3 ns
C/A	2	-48.5 ns	1.5 ns
B/C	1	-23.3 ns	1.7 ns
B/C	2	-22.4 ns	2.2 ns
C/B	1	+0.4 ns	1.9 ns
C/B	2	+0.8 ns	1.9 ns

TABLE III
SUMMARY OF CLOCK PULSE SYNCHRONIZATION RESULTS WITH MASTER/SLAVE-SPECIFIC RECIPROCITY CALIBRATION.

M/S	Test Number	$\mathrm{mean}[\epsilon[k]]$	$\mathrm{std}[\epsilon[k]]$
A/B	1	+0.4 ns	2.0 ns
A/B	2	+0.7 ns	2.1 ns
B/A	1	-0.3 ns	2.1 ns
B/A	2	-0.2 ns	2.2 ns
A/C	1	-0.3 ns	1.0 ns
A/C	2	-0.4 ns	1.3 ns
C/A	1	-0.3 ns	2.3 ns
C/A	2	-0.1 ns	1.9 ns
B/C	1	-1.6 ns	1.9 ns
B/C	2	-0.4 ns	3.2 ns
C/B	1	+1.1 ns	1.6 ns
C/B	2	+1.1 ns	1.9 ns

where

$$X_{\mathrm{M/S}} \approx \begin{cases} +48 \text{ ns} & \mathrm{A/B} \\ -48 \text{ ns} & \mathrm{B/A} \\ +37 \text{ ns} & \mathrm{A/C} \\ -37 \text{ ns} & \mathrm{C/A} \\ -11 \text{ ns} & \mathrm{B/C} \\ +11 \text{ ns} & \mathrm{C/B}. \end{cases}$$

Observe that these results are consistent, e.g., $X_{\mathrm{A/B}} \approx X_{\mathrm{A/C}} + X_{\mathrm{C/B}}$, and anti-reciprocal, e.g., $X_{\mathrm{A/B}} \approx -X_{\mathrm{B/A}}$. The anti-reciprocal nature suggests that the offsets are caused small master/slave-specific non-reciprocities in each test setup.

We performed an additional set of tests with master/slave-specific "reciprocity calibration". Specifically, the slave node implemented a fixed clock shift based on the results in Table II in each test according to the current master/slave configuration. The results of these tests are summarized in Table III. These results demonstrate that, with calibration of the non-reciprocal effects, the timestamp-free synchronization protocol with audio-frequency signaling can synchronize a pair of nodes realized by C6713 DSKs to within a few nanoseconds.

VI. CONCLUSIONS

This paper presented a real-time implementation of timestamp-free synchronization using audio-frequency signaling and Texas Instruments TMS320C6713 DSKs. Experimental results demonstrated that the slave node is able to predict the clock pulse observations from the master node to within a few nanoseconds over one second prediction intervals. Recordings of synchronized master and slave clock pulses showed that the nodes were able to synchronize to within a repeatable mean offset of less than 100 ns with standard deviations of approximately 1-3 ns. By compensating for the non-reciprocal effects in the system, the mean offsets were effectively canceled. Synchronization between two DSKs to within ± 4 ns was demonstrated for all tested configurations.

Source code for the master and slave nodes can be downloaded from https://github.com/anachronism/Timestamp-Free-Synchronization-For-TMS320C6713-DSK/tree/master.

REFERENCES

[1] D. Mills, "Internet time synchronization: the network time protocol," *IEEE Trans. Commun.*, vol. 39, no. 10, pp. 1482–1493, Oct. 1991.
[2] "IEEE 1588 Home Page," http://www.ieee1588.com/.
[3] W. Lewandowski, J. Azoubib, and W. Klepczynski, "GPS: primary tool for time transfer," *Proceedings of the IEEE*, vol. 87, no. 1, pp. 163–172, Jan 1999.
[4] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: a survey," *IEEE Netw.*, vol. 18, no. 4, pp. 45–50, Jul.–Aug. 2004.
[5] R. Mirollo and S. Strogatz, "Synchronization of pulse-coupled biological oscillators," *SIAM J. on Appl. Math.*, pp. 1645–1662, 1990.
[6] D. Lucarelli and I. Wang, "Decentralized synchronization protocols with nearest neighbor communication," in *Proceedings of the 2nd international conference on Embedded networked sensor systems (ACM SenSys'04)*, November 3-5 2004, pp. 62–68.
[7] Y.-W. Hong and A. Scaglione, "A scalable synchronization protocol for large scale sensor networks and its applications," *IEEE Journal on Selected Areas in Comm.*, vol. 23, no. 5, pp. 1085–1099, May 2005.
[8] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, and R. Nagpal, "Firefly-inspired sensor network synchronicity with realistic radio effects," in *Proceedings of the 3rd international conference on Embedded networked sensor systems (ACM SenSys'05)*, November 2-4 2005, pp. 142–153.
[9] A.-S. Hu and S. Servetto, "On the scalability of cooperative time synchronization in pulse-connected networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2725–2748, Jun. 2006.
[10] D.R. Brown III and A.G. Klein, "Precise timestamp-free network synchronization," in *Conf. Inf. Sciences and Systems (CISS2013)*, Mar. 2013.
[11] S. Ganeriwal, R. Kumar, and M. Srivastava, "Timing-sync protocol for sensor networks," in *Proceedings ACM SenSys 2003*. ACM New York, NY, USA, Nov. 2003, pp. 138–149.