Network Time Synchronization for Large Multi-hop Sensor Networks using the Cooperative Analog-and-Digital (CANDI) Protocol

Yong Jun Chang*, Haejoon Jung*, Sunghwan Cho†, and Mary Ann Weitnauer*
*School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, Georgia 30332-0250
†Korea Military Academy, Seoul, South Korea
Email: yongjun.chang@gatech.edu; hjung35@gatech.edu; scho@kma.ac.kr; mai@gatech.edu

***Abstract*—For multi-hop wireless sensor networks (WSNs), time synchronization (TS) across distributed nodes is important for certain applications (i.e., surveillance, gunshot location system, etc.). In this paper, a method combining two forms of cooperative transmission (CT), digital-based Concurrent Cooperative Transmission (CCT) and analog-based Semi-Cooperative Spectrum Fusion (SCSF), is proposed and demonstrated to achieve fast and accurate time synchronization over large multihop WSNs. CCT, considered in many physical layer system studies and realistically demonstrated, enables range extension. Range extension implies that the source timestamp can be diffused to the network faster (i.e., in fewer hops) than non-CT-based TS protocols. SCSF is an analog CT method where different cooperators transmit correlated information simultaneously. The two methods combine to create a new distributed method of network TS, called the Cooperative Analog and Digital (CANDI) TS protocol, which promises significantly shorter protocol time and smaller time errors in multi-hop networks. This paper presents the algorithm description, and physical layer simulations as well as network experimental results over a line network that show the performance advantage of CANDI over TPSN.**

I. Introduction

Wireless sensor networks (WSNs) have gained world-wide attention in recent years, particularly with the proliferation in micro electro-mechanical systems (MEMS) technology, which has facilitated the development of smart sensors [1]. WSNs consist of a large number of sensing nodes, which monitor various physical features, such as temperature, pressure, humidity, or location of object in a certain region. Using wireless communication, they transmit the collected information to the sink node [2].

WSNs need synchronized clocks over the whole network to determine the duty cycle and schedule tasks. In addition, many applications of WSNs require accurate time synchronization (TS) at each node so that data measured in different areas of the network can be properly time-tagged and later processed (e.g., correlated) at a central location. Some of these applications such as object tracking [3] and global structural health monitoring [4], require TS functionality to extract meaningful information from collected data, and involve large coverage areas.

Although WSNs are often distributed over large areas, the sensor radios have only a short range. Multi-hop communication can reduce the cost of such a network by not requiring that every sensor be within one hop of the higher-functioning and more expensive sink nodes. Multi-hop communication presents additional challenges to the typical wireless communication protocol. For example, synchronizing the clocks on all nodes in the network, when there is no external reference, is more difficult than for a star topology network [5]–[7]. While GPS can provide high-quality TS in some applications, the GPS signal is often not available at sensor locations, such as indoors, or costs too much energy. This paper explores a new approach, which uses cooperative transmission (CT) [2] range extension, to distribute the absolute time in two fast sweeps of the network.

Inspired by the high quality of CT time synchronization recently demonstrated [6], and by the low and stable synchronization error statistics in cascaded (multi-hop) CT [8], our proposed method exploits analog and digital forms of CT at different phases of the synchronization process. The first phase of the network uses concurrent cooperative transmission (CCT) [9], wherein cooperating nodes simultaneously transmit the same digital message in orthogonal channels that experience independent multi-path fading. Each receiver is capable of combining the differently faded copies, thereby achieving a significant SNR advantage, through array and diversity gains. In the analog phase, the cooperating nodes simultaneously transmit their individual estimates of the time, encoded across orthogonal dimensions. Nodes receiving this signal combat fading and reduce estimation error in one-step through the averaging inherent in diversity combining.

II. Related Work

Clock synchronization has been studied by many researchers for many years, and many TS protocols have been proposed so far. In centralized systems, each node can get the time by requesting the issue of time from the kernel, such as the Network Time Protocol (NTP) [10] server. However, in distributed systems like WSNs, there is no global timeserver. In the absence of GPS or some other external time reference, each node must have its local clock, and the distribution of time among the sensor nodes, i.e., a TS procedure, is necessary.

GPS-based TS is very accurate, but it is not appropriate for all WSNs because of its high cost, low energy efficiency, and limitation in indoor environments. The Precision Time Protocol (PTP) in [11] can achieve clock accuracy in the sub-microsecond, but it is suitable only for the hierarchical master-slave architecture [12].

Timing-sync Protocol for Sensor Networks (TPSN) [13] is a pair-wise communication based sender-receiver scheme. TPSN consists of two phases: level discovery and synchronization. In the level discovery phase, the sensor nodes identify their hop count levels from the root (time information source). In the synchronization phase, hop count is determined either by simple flooding or minimum spanning tree. In the synchronization phase, a Level $i+1$ node synchronizes its clock to a Level i node by exchanging time stamps, sequentially starting from the root to the following level hops. The synchronization process is based on two-way message exchange between a pair of nodes. TPSN can eliminate the propagation time error by handshaking of timestamps, and the send time error can also be reduced by time-stamping the packet at the MAC layer rather than at the application layer. The measurement data using Berkeley motes show about twice better performance than RBS in terms of the average timing accuracy [13].

Flooding Time Synchronization Protocol (FTSP) [14] is significantly more accurate than RBS and TPSN. FTSP dynamically elects a root node with the least identifier, which broadcasts its current time periodically using a tree structure. As TPSN, FTSP uses MAC-layer time-stamping to minimize the uncertainty in the message delay. Each node collects multiple data pairs (time stamps and arrival times), and uses linear regression for an accurate synchronization. Even though FTSP provides higher accuracy than RBS and TPSN, FTSP has an error that grows exponentially with the size of the network [14], since FTSP is an open-loop TS method. Also, its high accuracy level is kept by periodic synchronization message and complex computation, which takes a long time for FTSP to finish TS in large scale network [15].

Distinct from the other TS protocols, Pulse-coupled oscillator (PCO)-based TS protocols, which can be found in [16]–[18], and references therein, are fully decentralized, because these schemes do not require root or gateway for the global clock. Inspired by biological systems such as fireflies or cardiac pacemaker cells, the PCO-based algorithms provide *synchronicity*, which is the ability to organize simultaneous collective action across a WSN rather than *time synchronization*, which means nodes share a common notion of time [16]. However, because the *synchronicity* achieved through the PCO-based algorithms in the physical layer cannot give timing information mapped back onto a real-world clock, the PCO-based protocols in this category cannot be used for applications that require timing information [19].

In this paper, we propose a novel TS method based on cooperative transmission (CT), assuming a single root node. Even though the superposition of signal is exploited advantageously also in the PCO-based protocols [18], PCO-based range extension happens gradually because of its slow convergence especially in large WSNs. On the other hand, our proposed TS algorithm called Cooperative Analog and Digital (CANDI) takes full advantage of transmit diversity and range extension. Moreover, CANDI completes the entire process significantly faster by using both analog and digital forms of CT, which can synchronize a large WSN in fewer hops. The time synchronization of the digital phase (first phase) uses a "center of mass" method that is robust against carrier phase offsets and has been demonstrated on software-defined radios [9]. Furthermore, because all transmissions are between clusters and averaging is used in both phases, CANDI promises to be robust to node motion [20].

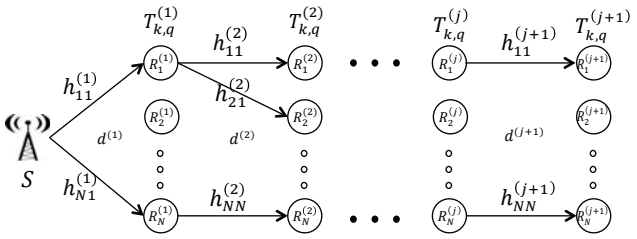

Fig. 1. Multi-hop distributed concurrent cooperative transmission.

III. Protocol Description

In this section, we present a high-level description of our proposed CANDI time synchronization protocol. Section IV discusses implementation details. The protocol uses a form of CT called the opportunistic large array (OLA). An OLA is a group of radios that relay the same packet, simultaneously, after receiving and correctly decoding the packet from a source or another OLA [21]. A "cluster" in this OLA context means a collection of nodes that have been opportunistically formed to have the same decoding level or CT-hop count in multi-hop OLA broadcasting [20]. More on how to form the cluster will be addressed in the following subsection. The distributed nature of an OLA creates spatial diversity, which provides range extension and reduces the hop count to reach the edge of the network [22], [23].

In this section, we use a superscript and a subscript to express the cluster and node index, respectively. For example, $R_k^{(j)}$ is Node k in the j^{th} cluster. As an example of a two-subscripts case, $h_{21}^{(j)}$ is the channel between Node 1 in the $(j-1)^{th}$ cluster and Node 2 in the j^{th} cluster. Similarly, $T_{21}^{(j)}$ is the time of that the beginning of a packet from Node 1 in the $(j-1)^{th}$ cluster reaches Node 2 in the j^{th} cluster. This time is referred to as the start-of-packet (SOP) time of the packet throughout this paper. $\hat{T}_{21}^{(j)}$ represents the estimated value of the SOP time $T_{21}^{(j)}$.

We analyze the CANDI protocol using the network topology shown in Fig. 1. We assume that by operating OLA transmission in the first phase, $N^{(j)}$ nodes form a cluster in the j^{th} hop as shown in the figure. In [6] it was shown that by using the center-of-mass method for SOP time estimation, the weighted average of estimated SOP times by channel gain is an approximated Best Linear Unbiased Estimator, which effectively minimizes the estimated time errors across the nodes in the receiver clusters, if the propagation time differences between nodes in the cluster are negligible. Therefore at Node k in the j^{th} cluster the combining coefficient $a_{kn}^{(j)}$ for estimated SOP

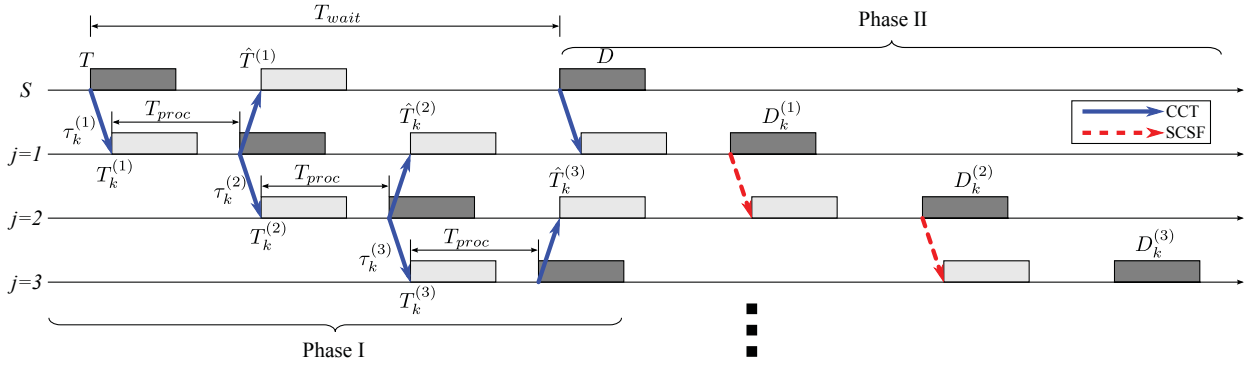

Fig. 2. CANDI network time synchronization protocol.

time $\hat{T}_{kn}^{(j)}$ becomes

$$a_{kn}^{(j)} = \frac{|h_{kn}^{(j)}|^2}{\sum_{i=1}^{N^{(j-1)}} |h_{ki}^{(j)}|^2}. \quad (1)$$

CANDI consists of two phases: 1) the root time broadcast and 2) the propagation time correction, as shown in Fig. 2, in which the horizontal axis indicates time and the vertical axis indicates a node or cluster index. In the first phase, the source node (or the root) broadcasts its timestamp to the network using OLA broadcasting. In the meantime, the nodes the $(i-1)^{th}$ cluster estimate and record a propagation time, e.g., $\hat{T}^{(i)}$, to the i^{th} cluster or node by overhearing the ongoing broadcast message. The source initiates the second phase after waiting some amount of time, T_{wait}, to avoid intra-flow interference [24] of the first phase broadcasting. The source broadcasts its estimation value of propagation delay to the first cluster's nodes. Once the nodes in the first cluster receive the source's estimated propagation delay information, they adjust their time based on the source timestamp and estimated propagation delay from the first and second phase respectively. The nodes in the first decoding level, then, rebroadcast their accumulated propagation delay to the next decoding level using Semi-Cooperative Spectrum Fusion (SCSF) [25], which is an analog cooperative transmission method. Next, we explain the phases in more detail.

A. Phase I: Root time broadcast

The source initiates the OLA broadcast with its timestamp (root time) embedded into the broadcast message at time T as shown in Fig. 2. The nodes in the first cluster estimate the SOP time of the source's packet. In this case, the transmitted signal is a transmission by just one radio, so there is no combining. The nodes in the first cluster decode the source packet and reset their local clock to match the SOP time to the embedded source's timestamp. Now the nodes schedule their rebroadcasting by adding a fixed processing time T_{proc} to the estimated SOP time $\hat{T}_k^{(1)}$. The time-to-live (TTL) field in the rebroadcasting message should be decremented by one. The rebroadcasting by Node k in the first cluster occurs at $\hat{T}_k^{(1)} + T_{proc}$. The source node overhears and estimates SOP time of this rebroadcasting message and calculates the propagation time from the source to the first decoding level by $\tilde{\tau}^{(1)} = (\tilde{T}^{(1)} - T - T_{proc})/2$. The source node records its estimated propagation delay for the second phase.

Node k in the j^{th} cluster receives $N^{(j-1)}$ copies of the source message from the previous cluster. The node estimates SOP times $\hat{T}_{kn}^{(j)}$ where $1 \leq n \leq N^{(j-1)}$, effectively using (1) to produce a combined SOP time. All nodes in the j^{th} cluster also rebroadcast the source message with decremented TTL using CCT. The following rebroadcasted messages from the $(j+1)^{th}$ cluster will be overheard by the nodes in the j^{th} cluster. For calculating the propagation delay, they also combine multiple SOP times using (1) to determine a combined SOP time. In this case, the estimated propagation delay at Node k in the j^{th} cluster becomes a weighted average of multiple propagation delays because of the combining method in (1). The broadcasting of the source's timestamp will continue until the TTL field in the broadcasting packet becomes zero.

B. Phase II: Propagation time correction

The source starts the second phase at $T + T_{wait}$, where T_{wait} is a pre-determined value, which should be at least $3 \cdot \big(T_{proc} + max(\tau)\big)$ to avoid the intra-flow interference [26]. The source node digitally encodes its estimated propagation delay $\hat{\tau}^{(1)}$ into the second phase's message. The nodes in the first cluster receive the source message and adjust their clock offsets by adding $\hat{\tau}^{(1)}$, which is decoded from the message.

The k^{th} node in the first cluster encodes its own notion of the accumulated propagation delay, $D_k^{(1)} = \hat{\tau}^{(1)} + \hat{\tau}_k^{(2)}$, where $\hat{\tau}^{(1)}$ is the received propagation delay information from the source and $\hat{\tau}_k^{(2)}$ is its own estimated propagation delay it got by overhearing the next cluster's rebroadcast messages in Phase I. Because each node has a slightly different message to transmit, that is, each node has a slightly different notion of accumulated two-hop propagation delay, CCT cannot be used. Therefore, we use SCSF. The k^{th} node encodes its $D_k^{(1)}$ using linear frequency modulation (FM), i.e., the k^{th} node transmits a single tone that is offset from the estimate of the received carrier frequency according to an offset $f_k^{(1)} = f_o + \alpha D_k^{(1)}$. Each node in the receiving cluster, which is Cluster 2 ($j = 2$), estimates the received spectrum, and uses the center-of-mass of that spectral estimate as the estimate of $f_k^{(1)}$. In general, Node k in the j^{th} cluster can decode the SCSF message from the $(j-1)^{th}$ cluster as

$$\tilde{D}_k^{(j-1)} = \frac{1}{\alpha}\left(\frac{\sum_{n=1}^{N_{win}} v_n |\mathcal{F}_k(n)|^2}{\sum_{n=1}^{N_{win}} |\mathcal{F}_k(n)|^2}\right), \quad (2)$$

where N_{win} is FFT size, v_n is the n^{th} DFT frequency, $\mathcal{F}_k(n)$ is n^{th} DFT value of the received baseband SCSF signal [25]. The node adjusts its local clock by adding the decoded accumulated propagation delay $\tilde{D}_k^{(j-1)}$. The node next retransmits the reencoded SCSF message, $D_k^{(j)} = \tilde{D}_k^{(j-1)} + \hat{\tau}_k^{(j)}$, to the next cluster.

In the second phase, the nodes in each cluster stop their synchronization operation right after transmitting an accumulated propagation delay to the next cluster. Therefore, overhearing operation does not necessarily occur in the second phase. The transmission will also continue until the embedded TTL field becomes zero.

IV. Implementation Details

Since both CCT and SCSF are not supported by off-the-shelf radios, we implement our proposed algorithm using USRP1 and GNU Radio Software-Defined Radio (SDR). GNU Radio is used as a base-band signal-processing module, which operates on a PC that is attached to the USRP1 via the USB interface [27] [28]. GNU Radio is an open-source software package that is usually run on a Linux operating system. It has various signal processing blocks written in the C++ language and protocol stacks written in the Python language. For the first phase of CANDI, we used the same physical layer and the protocol stack, as that used in [29].

A. SCSF Implementation

Unlike the SCSF application in [25], which is aerial reading of a sensor field in a LOS channel, we consider a fading channel environment. Also, different from [25], we consider imperfection of frequency and timing recovery in the SCSF decoding process. Each transmitter converts its desired data into frequency using a modulation index. The receiver first detects the start of SCSF-encoded subframe using BFSK-encoded preamble. The subframe passes through the FFT and mag-square blocks for non-coherent decoding, in which the mag-square output of FFT is the periodogram. The SCSF receiver finally calculates the center-of-mass of the periodogram to estimate the average of transmitted data. The SCSF algorithms can be realized as follows.

1) Modulation of Data: Node k converts its desired data s_k scaled by the modulation index ζ into frequency. The modulation index is chosen in such that $|s_k \cdot \zeta| < 1/N_{win}$ for all s_k. Discrete baseband representation of the transmitted SCSF signal from node k can be written as

$$x_k[n] = e^{j2\pi s_k \zeta n}, \quad (3)$$

where $n = 0, \ldots, N_{win} - 1$.

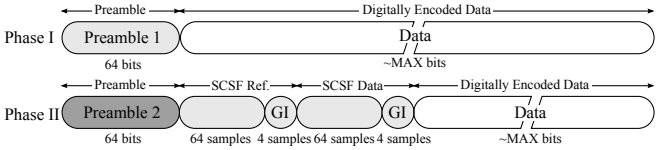

Fig. 3. Frame structures for CANDI algorithm.

2) Imperfection of Carrier Frequency Recovery: Since SCSF modulates its data into frequency, the discrepancy of the receiver local clock oscillator, which provides a reference clock to the carrier frequency mixer, causes a misreading of SCSF encoded data. At the receiver, the superimposed signal from N transmitters with carrier frequency offset $\triangle_k$ can be written as

$$\begin{aligned} y[n] &= \sum_{k=1}^{N} h_k \cdot e^{j2\pi s_n \zeta n} \cdot e^{j2\pi \frac{\triangle_n}{f_s} n} + \omega[n] \\ &= \sum_{k=1}^{N} h_k \cdot e^{j2\pi (s_n \zeta + \frac{\triangle_k}{f_s}) n} + \omega[n], \end{aligned} \quad (4)$$

where f_s is the sampling frequency. The FFT of $y[k]$ can be calculated by

$$\begin{aligned} \mathcal{F}[m] &= \sum_{l=0}^{N_{win}-1} \left\{ \left(\sum_{k=1}^{N} h_k e^{j2\pi (s_n \zeta + \frac{\triangle_k}{f_s}) l} \right) + \omega[l] \right\} e^{-j2\pi \frac{m}{M} l} \\ &= \sum_{l=0}^{N_{win}-1} \left(\sum_{k=1}^{N} h_k e^{j2\pi (s_n \zeta - \frac{k}{M} + \frac{\triangle_n}{f_s}) l} \right) + \Omega[m], \end{aligned} \quad (5)$$

where $\Omega[m] = \sum_{l=0}^{N_{win}-1} \omega[l] e^{-j2\pi \frac{m}{M} l}$. If we assume the unit channels $h_k = 1$ for all k and the noise free channel, the estimated SCSF value of (5) will indicate the average value of the net or superimposed data, which is the sum of designated data and frequency offset normalized by $f_s \zeta$,

$$\tilde{D}_{net} \approx \frac{1}{N} \sum_{k=1}^{N} \left(s_k + \frac{\triangle_k}{f_s \zeta} \right). \quad (6)$$

If we set the transmit data s_k to zero for all index k, only the frequency offset term in (6) is left. Therefore, by transmitting the reference SCSF frame that encodes zero as a data along with the actual SCSF frame, the SCSF-encoded data with a frequency offset can be estimated by simply taking the subtraction the reference from the net as

$$\tilde{D} = \tilde{D}_{net} - \tilde{D}_{ref}. \quad (7)$$

Fig. 3 shows the frame structures for the first and second phase of CANDI. The frame structure in the first phase is a typical frame structure for a digital communication with a preamble. In contrast, the second phase of CANDI has two sets of SCSF subframes, in which the first subframe is the reference SCSF subframe to send a zero-encoded data while the second subframe sends the actual data (i.e., $D_k^{(i)}$).

3) Imperfection of Transmit Time: Not only frequency offsets between nodes but also transmit timing errors can affect the decoding process of SCSF. At the receiver side, a number of SCSF frames are superimposed, in which each frame has a slightly different timing. We insert a guard interval into the

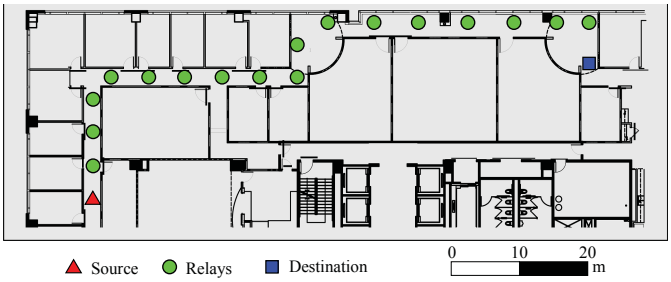

Fig. 4. Floor plan and node placement for the experiment.

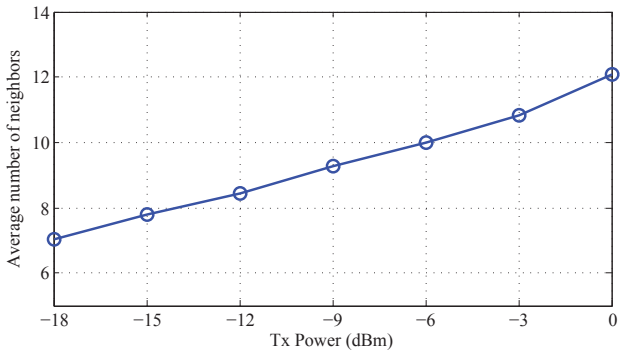

Fig. 5. Average number of neighbors in the two topologies.

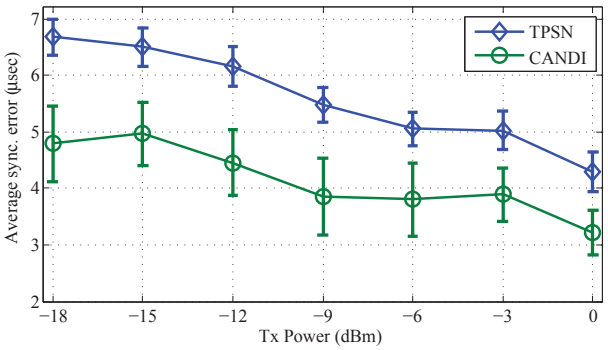

Fig. 6. RMS timing error between extreme nodes.

end of each SCSF subframe, like the OFDM guard interval, to compensate for the multipath delays and transmit time offsets. The content of the guard interval is a copy of the first part of the SCSF subframe. Once the receiver determines the starting point of the SCSF subframe based on the preamble detection, N_{win} samples can be taken with $|GI|/2$ offset where $|GI|$ is the length of the guard interval.

V. Experimental Setup and Results

We present the experimental design and the results of the proposed protocol, CANDI, and the comparison target, TPSN. The experiment is conducted using 19 nodes, which were deployed on the fifth floor of the Centergy building of the Georgia Institute of Technology, as shown in Fig. 4. The nodes are placed to create an equidistant linear network along the hallways.

In case of a global synchronization for a large-scale multi-hop network, it is likely that the timing error of a certain TS protocol increases as the hop count (or the latency) from the root node increases. For larger networks, the node with the largest hop distance would suffer from the biggest error by the clock drift that is not compensated in TPSN. Therefore, the error between the root and the final node with largest hop distance can demonstrate the performance and suitability in a large-scale network, which is the case of interest in this paper.

However, it is practically difficult to measure the timing error of the two nodes physically far apart, accurately. Therefore, we design the measurements in a round trip manner. To be specific, the source and destination nodes are the two ends of the two-way synchronization, which consists of forward- and reverse-path synchronizations. For the forward path, the other nodes in the network synchronize to the source node. If the destination node is successfully synchronized to the source node after the forward path synchronization, we start the reverse path synchronization by immediately re-initialization of the network-time protocol algorithm using the destination as the root node. When the source node is also successfully synchronized to the destination after the reverse path synchronization, the initial source node calculates the time gap (timing error) between the initial clock and the synchronized clock from the reverse path as our performance metric, which is also easy to measure because we extract the two timing information in the same node (the source) without introducing additional errors or distortions in the measurement.

For orthogonal sub-channel assignment in a CANDI protocol, we manually assign one of the four diversity channels to each node in incremental fashion (i.e., {1,2,3,4,1,2,3,...} from end to end). Even though manual assignment can be replaced by a distributed algorithm, the design of such algorithm is beyond the scope of this paper.

We consider different transmit power levels by increasing from -18dBm to 0dBm with 3dBm intervals (total of 7 different power levels). Higher transmit power increases the node degree, which means the average number of one hop neighbors as defined in [20], of the network as shown in Fig. 5, thereby decreasing hop count from the source to the destination.

Fig. 6 shows the root mean square (rms) synchronization error after the two-way phase, which is averaged over 200 trials. In the figure, the blue curve with the 'd'-markers and the green curve with the 'o'-markers represent TPSN and CANDI, respectively. The 'I-shaped' bars indicate the 95% confidence intervals of the statistical data. If we look at the trend of the two curves under the change in transmit power, we can observe that the timing errors of the both protocols decrease as the transmit power increases. We consider two possible reasons for this: the SOP time estimation error and the clock drift error. First, the SOP time error is inversely proportional to the received SNR, so the higher transmit power reduces the SOP time errors for the both algorithms. Second, as transmit power increases, the hop count from one end to the other end decreases as shown in Fig. 7. Also, this hop count saving by the increased transmit power level cuts down the overall latency to finish the two-way phase of the two protocols. Because the clock drift error is proportional to the latency, TPSN and CANDI will be less distorted by the clock

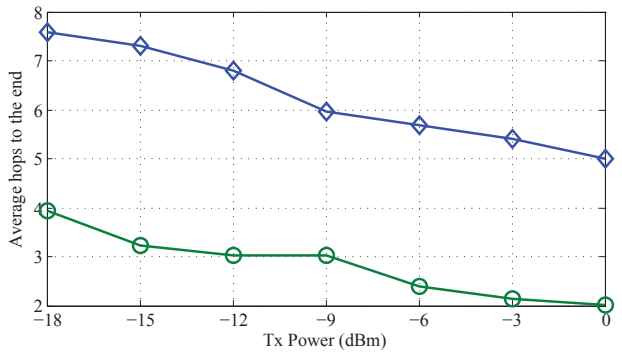

Fig. 7. Average hop count.

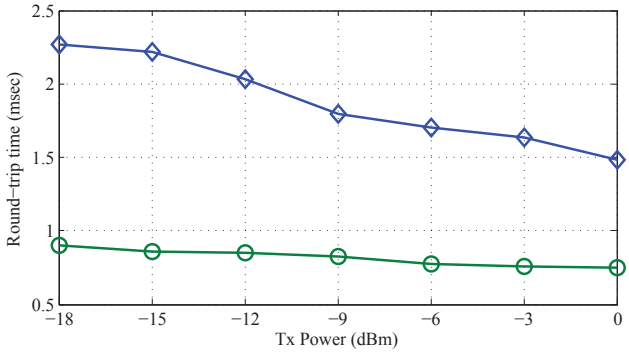

Fig. 8. Average round trip time.

drift as the transmit power increases.

A more important observation is that CANDI outperforms TPSN across all transmit power levels; specifically (as shown in Fig. 6), CANDI provides reductions in the rms timing error, relative to TPSN, of 22.6% at -3dBm up to 29.5% at -18dBm, which can be explained as follows. First, CANDI achieves SNR advantage though array and diversity gains of CCT in the SOP detection, which makes the SOP estimation error smaller compared to TPSN based on the conventional SISO links. Second, the range extension property of CT allows hop count saving of CANDI as shown in Fig. 7 over the all transmit power ranges, which makes CANDI take less time to finish the two-way phase as in Fig. 8. Therefore, compared to TPSN, CANDI suffers less from the clock drift.

VI. Conclusion

In this paper, two forms of cooperative transmission (CT), digital-based Concurrent Cooperative Transmission (CCT) and analog-based Semi-Cooperative Spectrum Fusion (SCSF), are combined to create a new distributed method of network time synchronization (NTS) protocol to achieve fast and accurate time synchronization over large multi-hop wireless sensor networks (WSN)s. The experimental results show that the averaging and range extension benefits of CCT lead to reductions in worst case rms timing errors of up to 29.5%.

References

[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *ELSEVIER Computer Networks*, vol. 52, no. 12, pp. 2292 – 2330, 2008.

[2] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Trans. on Infor. Theory*, vol. 50, no. 12, pp. 3062–3080, Dec. 2004.

[3] C.-Y. Lin, W.-C. Peng, and Y.-C. Tseng, "Efficient in-network moving object tracking in wireless sensor networks," *IEEE Trans. on Mobile Computing*, vol. 5, no. 8, pp. 1044 –1056, Aug. 2006.

[4] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, "Health monitoring of civil infrastructures using wireless sensor networks," *in Proc. 6th Int. Sym. on IPSN*, pp. 254 –263, Apr. 2007.

[5] I. F. Akyildiz and M. C. Vuran, "Wireless sensor networks," 2010.

[6] Y. J. Chang and M. A. Ingram, "Convergence property of transmit time pre-synchronization for concurrent cooperative communication," *IEEE GlobeCom*, Dec. 2010.

[7] B. Sklar, *Digital communications: fundamentals and applications*. Prentice-Hall, Inc., 1988.

[8] Z. Gao, Y. J. Chang, and M. Ingram, "Synchronization for cascaded distributed MIMO communications," *in Proc. IEEE MILCOM*, pp. 387 –392, 3 2010.

[9] Y. J. Chang and M. Ingram, "Convergence property of transmit time pre-synchronization for concurrent cooperative communication," *in Proc. IEEE GLOBECOM*, pp. 1 –5, Dec. 2010.

[10] D. Mills, "Internet time synchronization: the network time protocol," *IEEE Trans. on Comm.*, vol. 39, no. 10, pp. 1482 –1493, Oct. 1991.

[11] "IEEE standard for a precision clock synchronization protocol for networked measurement and control systems," *IEEE Std. 1588-2008*, 2008.

[12] M. Akhlaq and T. Sheltami, "RTSP: An accurate and energy-efficient protocol for clock synchronization in WSNs," *IEEE Trans. on Instrumentation and Measurement*, vol. 62, no. 3, pp. 578–589, 2013.

[13] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," *in Proc. 1st Intl. Conf. on Embedded Networked Sensor Systems*, pp. 138–149, 2003.

[14] M. Maróti, B. Kusy, G. Simon, and A. Lédeczi, "The flooding time synchronization protocol," *2nd Intl. Conf. on Embedded networked sensor systems*, pp. 39–49, 2004.

[15] L. Sun, H. Wang, T. Yan, and J. Liu, "ESTS: An error statistic based time synchronization protocol for wireless sensor networks," *Springer Computational Intelligence*, vol. 4114, pp. 1317–1323, 2006.

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

[17] Y. Wang and F. Doyle, "Optimal phase response functions for fast pulse-coupled synchronization in wireless sensor networks," *IEEE Trans. on Signal Processing*, vol. 60, no. 10, pp. 5583–5588, 2012.

[18] R. Pagliari and A. Scaglione, "Scalable network synchronization with pulse-coupled oscillators," *IEEE Trans. on Mobile Computing*, vol. 10, no. 3, pp. 392–405, 2011.

[19] M. Leng and Y.-C. Wu, "Distributed clock synchronization for wireless sensor networks using belief propagation," *IEEE Trans. on Signal Processing*, vol. 59, no. 11, pp. 5404–5414, 2011.

[20] L. Thanayankizil, A. Kailas, and M. A. Ingram, "Routing for wireless sensor networks with an opportunistic large array (OLA) physical layer," *Ad Hoc & Sensor Wireless Networks*, vol. 8, no. 1-2, pp. 79–117, 2009.

[21] A. Scaglione and Y.-W. Hong, "Opportunistic large arrays: cooperative transmission in wireless multihop ad hoc networks to reach far distances," *IEEE Trans. Sig. Proc.*, vol. 51, no. 8, pp. 2082 – 2092, Aug. 2003.

[22] Y. J. Chang, H. Jung, and M. A. Ingram, "Demonstration of a new degree of freedom in wireless routing: Concurrent cooperative transmission," *HotEMNETS 2010 Workshop on Hot Topics in Embedded Networked Sensors*, June 2010.

[23] H. Jung, Y. Chang, and M. A. Ingram, "Experimental range extension of concurrent cooperative transmission in indoor environments at 2.4GHz," *in Proc. IEEE MILCOM*, Oct. 2010.

[24] H. Jung and M. A. Ingram, "Analysis of intra-flow interference in opportunistic large array transmission for strip networks," *in Proc. IEEE ICC*, June 2012.

[25] A. Akanser and M. A. Ingram, "Semi-cooperative spectrum fusion (SCSF) for aerial reading of a correlated sensor field," *International Conference on Wireless Comm., VITAE*, pp. 732 –736, May 2009.

[26] A. Baghaie and B. Krishnamachari, "Fast flooding using cooperative transmissions in wireless networks," *in Proc. IEEE ICC*, pp. 1–5, June 2009.

[27] "GNU Radio," http://gnuradio.org.

[28] Ettus, "The universal software radio peripheral," Website, http://www.ettus.com/downloads/er_broch_trifold_v5b.pdf.

[29] Y. J. Chang, H. Jung, and M. A. Ingram, "Demonstration of an OLA-based cooperative routing protocol in an indoor environment," *European Wireless Conference*, pp. 1 –8, Apr. 2011.