ROBUST TIME-SLOTTED ROUND-TRIP CARRIER AND TIMING SYNCHRONIZATION FOR DISTRIBUTED BEAMFORMING

Martí Mañosas-Caballú, Gonzalo Seco-Granados

Telecommunications and Systems Engineering Department, Universitat Autònoma de Barcelona
Edifici Q - Campus de la UAB, 08193, Bellaterra (Barcelona), Spain
phone: (+34) 93 581 40 30 , fax: (+34) 93 581 40 31, email: {marti.manosas, gonzalo.seco}@uab.cat
web: spcomnav.uab.es

ABSTRACT

Distributed beamforming has arisen as a significant approach in multi-user wireless communication systems. It allows several transmitters with common information to emulate an antenna array and focus their transmissions towards an intended destination. However, carrier and timing synchronization among the transmitters is necessary to ensure that the information is aimed in the desired direction. In this work we present a robust time-slotted round-trip carrier and timing synchronization protocol that is valid in dynamic environments in the sense that sensors can disappear from the network without affecting the performance severely. The protocol is based on the execution of simple rules at each sensor and leaves freedom to choose those signals that provide better timing synchronization. Numerical results show that a large fraction of the maximum beamforming gain can be maintained for death ratios as high as 50%.

1. INTRODUCTION

A fundamental problem in ad-hoc wireless networks such as wireless sensor networks (WSN) is the use of energy-efficient communication techniques. Lately, much of the research in this area has focused on cooperative approaches. The term “cooperative communication” typically refers to a system where users share and coordinate their resources in order to improve the quality of their transmissions [1]. The idea is particularly attractive in wireless scenarios due to both the large variety of channel qualities for different transmitter-receiver pairs and the limited energy and bandwidth resources.

Conventional transmit beamforming is a communication technique that allows a transmitter with several antennas to focus its bandpass signal in an intended direction. The advantages of conventional transmit beamforming are numerous and well-documented in the literature [2]. For instance, by focusing the transmission towards the intended destination, less transmit power is needed to achieve a desired signal-to-noise ratio (SNR) target, which is known as *beamforming gain*. This feature is particularly appealing in wireless communication systems with energy constrained nodes such as sensor networks. However, in this type of systems nodes are typically too small to allow for the use of conventional antenna arrays.

Recently, the idea of transmit beamforming has been extended to distributed networks of single-antenna transmitters. By means of cooperation, transmitters can emulate a conventional beamformer and behave as a “distributed beamformer”. The idea is that individual sources with common information transmit with phase and time aligned carriers such that their bandpass transmissions combine constructively at the intended destination. Nevertheless, unlike conventional beamforming, in distributed networks each transmitter has an independent and imperfect oscillator. For this reason, it is necessary to synchronize the signals of the transmitters.

In this work we distinguish three types of synchronization procedures that complement each other for the pursued goal: *carrier frequency synchronization* lets each user work with the same carrier frequency; *carrier phase synchronization* makes each carrier arrive with the same phase at the destination; finally *timing synchronization* is necessary in order to achieve simultaneity of the signals at the receiver. Although there exist several papers that investigate the practical problem of multi-user carrier synchronization for distributed beamforming, most of them are still very preliminary because they either do not solve all types of synchronization or consider ideal scenarios. Some of these methods are [3–7].

One of the latest techniques can be found in [8], and it solves the three commented synchronization procedures while avoids the communication from sensors to destination. The other methods only focus on achieving frequency and phase synchronization, what implies that it can exist a mismatch in the symbols alignment at the receiver and hence an imperfect addition of the information is likely to occur. The impact of this phenomenon worsens for high data-rate transmissions. For this reason those methods that do not provide timing synchronization present limitations in many scenarios. In addition, it is also important to bear in mind that sensors in a WSN can leave the network without previous warning. An example of this happens when a sensor breaks down, hence being unable to send any type of notification prior to its disappearance. To the best of our knowledge, it does not exist any work that presents a distributed beamforming scheme suitable for this kind of dynamic environments.

In this paper we present a new synchronization protocol based on the work of [8]. The two main contributions of the paper are: i) the protocol is robust in dynamic environments in the sense that sensor disappearances can occur without affecting the performance of the system severely; ii) synchronization procedures are separated in such a manner that they allow the use of specific signals for high accuracy timing synchronization. The robustness of the protocol is achieved thanks to the execution of simple rules at each sensor.

We also analyse the performance of the proposed algorithm in terms of the beamforming gain and its dependency on estimation errors. We show that a large fraction of the maximum beamforming gain can be sustained, even when 50% of the existing nodes disappear during the synchronization procedure (i.e. when the death ratio is equal to 50%).

2. SYSTEM MODEL

Let us consider a system of N sensors S_j for $j = 1,\ldots,N$ arbitrarily distributed over some region and with a limited maximum transmit power. The index j represents only a logical position used to apply the different steps of the protocol described in Section 3.

Consider also a distant base station which acts as a destination (D_0) and is not power constrained. The nodes want to send a common message $m(t)$ modulated on a carrier frequency f_c to D_0, as shown in Fig. 1. In order to assure correct transmission of the information from the N sensors to the base station, a distributed beamforming protocol is applied. Note that the proposed model is not related to any specific network topology. In fact, it can represent one link of a larger network that includes many-to-one transmissions. For instance, $S_1,\ldots,S_N$ could be a group of relay nodes used to transmit from a distant source to the destination.

We suppose the transmitted carrier signal by node S_j is:

$$x_j(t) = \cos\left(2\pi f_j \left(t - t_j^*\right) + \phi_j\right) \quad t \geq t_j^* \tag{1}$$

where t refers to the time, f_j is the carrier frequency, ϕ_j is the initial phase and t_j^* is the transmission starting instant.

The channel from node S_i to node S_j (including D_0 with the index 0) is assumed to be *flat fading* and invariant with impulse response $h_{i,j}$ during the transmission of a message. Therefore, the channel can be characterized by an attenuation $\alpha_{i,j}$, a propagation delay $\tau_{i,j}$, and a phase $\phi_{i,j}$ on the top of the one caused by the propagation delay. In this situation the signals will combine constructively at the destination whenever the carriers are fully synchronized. Thus we can simplify the problem by omitting each baseband signal and considering only the corresponding carrier. Finally, we also assume channel reciprocity such that $h_{i,j} = h_{j,i}$.

With these assumptions, the received signal at destination D_0 produced by the transmission of the signal (1) is:

$$y_{j,0}(t) = \alpha_{j,0} \cos\left(2\pi f_j t + \phi_{j,0}^{eq}\right) + n_0(t) \tag{2}$$

where $\phi_{j,0}^{eq} = \phi_j + \phi_{j,0} - 2\pi f_j t_j^* - 2\pi f_j \tau_{j,0}$ is the equivalent phase of the signal at time $t = 0$ and $n_0(t)$ denotes the additive white Gaussian noise (AWGN). Note that the expression of (2) is only valid for $t \geq t_j^* + \tau_{j,0}$. This formulation will help us to explain our new round-trip protocol in section 3.

Next we describe the key aspects that make distributed beamforming more challenging than the conventional case. First of all we remark that each source keeps its local time using its own local oscillator, which implies the following:

- The nodes do not possess a common time scale due to the use of imperfect oscillators.
- The nodes do not possess a common time reference due to the use of independent oscillators.

Because of this, none of the nodes in the system know the "true" time and then they do not know the "true" frequency or phase of their local oscillator either. This means that nodes cannot generate absolute phase or frequency estimates with respect to the "true" time. We model this phenomenon by relating the "true" time t and the j-th sensor time $t^{(j)}$ through a multiplicative factor ε_j and an offset δ_j:

$$t^{(j)} = \varepsilon_j \cdot t + \delta_j \tag{3}$$

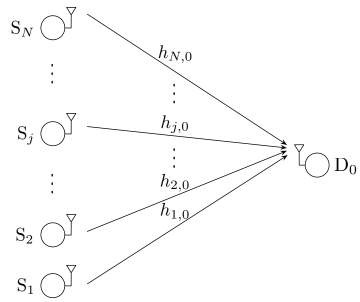

Figure 1: System model for a network of N sensors and a base station D_0.

In addition to the aforementioned two points, positions of the nodes are completely unknown either absolutely or relatively, which renders the delays $\tau_{j,0}$ between each sensor node and the base station unknown. The lack of a common time scale makes only coherence (i.e. carrier phase alignment) difficult and can be solved by means of frequency synchronization. However, the uncertainty of the positions and the lack of a common time reference make it difficult to obtain both simultaneity and coherence at the destination. This can be solved through phase and timing synchronization.

Finally note that the coherent transmission cannot last an unlimited period of time due to synchronization errors. The beamforming protocol will need to resynchronize periodically in order to avoid unacceptable phase drift during beamforming. As a consequence, we distinguish two separate procedures: the beamforming stage T_{beam} and the synchronization stage T_{syn}.

3. ROBUST ROUND-TRIP SYNCHRONIZATION PROTOCOL

In this section we present the proposed robust time-slotted round-trip carrier synchronization protocol. The main goal of the protocol is to achieve frequency, phase and timing synchronization even if several sensors disappear at any time without previous warning.

3.1 Protocol overview

Our key proposal lies in splitting the whole problem in three independent requirements at the destination and provide a specific solution to each of them. This let us make synchronization procedures simple and somehow independent.

The first requirement is that all carriers arrive simultaneously at destination. The second requirement is that all carriers arrive with the same initial phase at destination (phase of (2) at $t = t_j^* + \tau_{j,0}$). Finally, the third requirement is that all carriers arrive with the same frequency at destination. Note that the fulfilment of the first two requirements implies that carriers arrive at destination coherently at the start of the beamforming. On the other hand, the third requirement lets each user maintain coherence over time.

The synchronization protocol begins with a calibration step in which the destination broadcasts at t_c a sinusoidal tone $x_0(t)$ of duration T_c and frequency f_c to all sensors:

$$x_0(t) = \cos\left(2\pi f_c (t - t_c) + \phi_c\right) \quad t \in [t_c, t_c + T_c)$$

Then, each sensor locally estimates the phase and frequency of the received tone as detailed in subsections 3.3 and 3.4.

Afterwards a total of $2N-1$ consecutive timeslots TS_k for $k=0,\ldots,2N-2$ are used for the key part of the synchronization protocol, each of them starting at time t_k. Finally, an additional time T_{beam} is dedicated to beamforming. The activity in each timeslot is summarized here:

1. TS_0: the destination broadcasts a temporal reference signal $r_0(t)$ of duration T_0 to all sensors.
2. TS_k for $k=1,\ldots,N-1$: S_k broadcasts a temporal reference signal $r_k(t)$ of duration T_k to all sensors. The transmitted signal is sent just after the reception of the signal corresponding to the timeslot TS_{k-1}. We denote this set of slots as *up-cycle*.
3. TS_k for $k=N,\ldots,2N-2$: S_{2N-k} broadcasts a temporal reference signal $r_{2N-k}(t)$ of duration T_{2N-k} to all sensors. The transmitted signal is sent just after the reception of the signal corresponding to the timeslot TS_{k-1}. We denote this set of slots as *down-cycle*.
4. $\mathrm{TS}_{2N-1}=T_{beam}$: all sensors transmit to the destination as a distributed beamformer. This interval starts at t_{2N-1}, just after node S_1 receives the temporal reference signal from S_2, at the end of the down-cycle.

In addition to these basic rules, each node has to count the time that elapses since the reception of each temporal reference signal. When this time exceeds the value of $(2T_p)m$ for some $m\in\mathbb{N}$, each node must deduce that m consecutive nodes have disappeared. Here T_p is an upper bound on the propagation delay between any two nodes of the network, and it should be known by all sensors. The value of $2T_p$ guarantees that there is enough time for a sensor to receive the temporal reference signal that is sent at a given timeslot TS_k since the reception of the temporal reference signal that is sent at TS_{k-1}.

Thanks to the simple exchange of reference signals, each sensor can readily deduce which nodes have disappeared by means of a counter and update its network view. Specifically a given sensor S_j is supposed to update the total number of nodes to $N-m$ and its position to $j-m$ when m consecutive nodes with a logical position below j die. When m nodes with a logical position above the given sensor die, only the variable N should be updated. The node following the last dead one is the responsible for sending the next temporal reference signal. This simple procedure prevents the nodes from being blocked by waiting for the reception of signals from missing sensors. In the following we describe in more detail how each type of synchronization is achieved in our protocol.

3.2 Timing synchronization

Our timing synchronization proposal makes use of the whole set of $2N-1$ first timeslots. The rules are focused on letting each user know the precise moment it has to send the information towards the destination for achieving simultaneity during T_{beam}. In a general form each user S_j will send (at time t_j^*) after waiting a delay of τ_j seconds since the end of the emission of its temporal reference signal during the down-cycle:

$$t_j^* = t_{2N-j} + T_j + \tau_j \quad \forall\, j\in\{1,\ldots,N\} \tag{4}$$

Although the first node is not supposed to send any temporal reference signal in the down-cycle, it also has to send towards destination according to (4), at time $t_1^* = t_{2N-1} + T_1 + \tau_1$.

The key point here is the way each sensor deduces its own delay τ_j. Similar to [8], the calculation of τ_j results

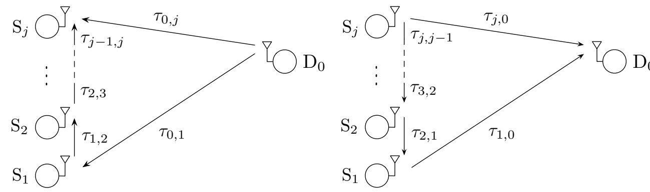

Figure 2: Path delays used for the deduction of τ_j.

from forcing that each signal $x_j(t)$ for $j=2,\ldots,N$ arrives at destination simultaneously with $x_1(t)$. A solution would be to use a value of τ_j equal to the difference between the arrival times at D_0 of both signals $x_j(t)$ and $x_1(t)$ when considering $\tau_j=0$ in (4). After assuming reciprocity at each channel, it is easy to proof that this difference time can be estimated from the elapsed time between the end of the observation times at S_j of the signal coming from S_{j-1} during the up-cycle and the signal coming from D_0:

$$\tau_j = \left(t_{j-1} + \tau_{j-1,j} + T_{j-1}\right) - \left(t_0 + \tau_{0,j} + T_0\right) \tag{5}$$

Fig. 2 depicts this idea. First, it shows the delays used by sensor S_j to calculate τ_j, and second the delays that S_j actually wants to emulate in order to achieve simultaneity.

However, in order to take into account possible disappearances supplementary rules must be applied. A node having m consecutive deaths below its logical position during the up-cycle, has an additional error in the calculus of τ_j corresponding to the performed waiting $(2T_p)m$ of the first alive sensor, which we denote by S_d. This waiting becomes an error because it is not performed again during the down-cycle.

In order to counteract this error, we propose that each affected node subtracts the known quantity $(2T_p)m$. Note that the generated time of a given sensor S_j will be affected by ε_j due to its imperfect clock (see (3)). Hence with this rule each node will produce $\overline{\tau_j}$ instead of (5) and use it in (4):

$$\overline{\tau_j} = \tau_j - \frac{(2T_p)\,m}{\varepsilon_j}$$

where the error in the correction is:

$$e_j = (2T_p)\,m\left(\frac{1}{\varepsilon_d} - \frac{1}{\varepsilon_j}\right) \tag{6}$$

In case that m deaths occur during the down-cycle, there is no way to correct τ_j, and it is appropriate that those nodes that are positioned below the dead ones cancel beamforming.

Algorithm 1 shows the commented rules in an algorithmic form. We emphasize that it corresponds to a reduced version of our implemented code and that a realistic algorithm must consider that nodes may face some critical situations.

3.3 Frequency synchronization

Let us consider that the sinusoidal tone received at sensor S_j in the absence of noise produced by the emission of $x_0(t)$ during the calibration step can be written as:

$$y_{0,j}(t) = \alpha_{0,j}\cos\left(2\pi f_c\left(t - t_c - \tau_{0,j}\right) + \phi_c + \phi_{0,j}\right) \tag{7}$$

which is only valid for $t\in\left[t_c+\tau_{0,j},\, t_c+\tau_{0,j}+T_c\right)$. As in (2), $\alpha_{0,j}$ and $\phi_{0,j}$ account for the channel effects, including multipath.

Algorithm 1 Network updating for timing synchronization

if m deaths below **then**
  if up-cycle **then**
    $N \leftarrow N-m$, $j \leftarrow j-m$, $\overline{\tau_j} = \tau_j - \frac{(2T_p)m}{\varepsilon_j}$
  else {down-cycle}
    $N \leftarrow N-m$, $j \leftarrow j-m$
  end if
else if m deaths above **then**
  if up-cycle **then**
    $N \leftarrow N-m$
  else {down-cycle}
    $N \leftarrow N-m$, cancel beamforming
  end if
end if

Then we propose that each sensor generates its local estimate $\hat{f}_{cj}$ of the frequency f_c from the received signal $y_{0,j}(t)$, and uses this estimation as the carrier frequency f_j of the signal (1) that has to be sent to the base station. Note that the frequency reading will be slightly modified by the time scale of each sensor, but the error will be counteracted by the same time scale when f_j is generated during T_{beam}.

3.4 Phase synchronization

In order to perform phase synchronization we must achieve that $\phi_{j,0}^{eq} = \phi_{i,0}^{eq}$ $\forall j,i \in \{1,\ldots,N\}$. Assuming that simultaneity is fulfilled at the destination, it is verified that $t_j^* + \tau_{j,0} = t_i^* + \tau_{i,0}$ $\forall j,i \in \{1,\ldots,N\}$. Hence, the previous condition reduces to:

$$\phi_j + \phi_{j,0} = \phi_i + \phi_{i,0} \quad \forall j,i \in \{1,\ldots,N\} \tag{8}$$

which is equivalent to what is explained at the beginning of section 3: force all carriers to arrive with the same initial phase at destination. Note that we have assumed in (8) that $f_j = f_i$ $\forall j,i \in \{1,\ldots,N\}$, which is reasonable as sensors will be frequency synchronized before sending towards D_0.

From (8) the only degrees of freedom are the initial phases of the transmitting carrier signals: ϕ_j for $j = 1,\ldots,N$. Thus the optimal solution is that each sensor S_j uses a value for its initial carrier phase equal to $-\phi_{j,0}$ when transmitting towards the destination. In the following we show how each sensor can get a good alternative to this value from the calibration step.

Denoting the initial phase of $y_{0,j}(t)$ as $\varphi_{0,j}$, that corresponds to the phase of (7) at time $t = t_c + \tau_{0,j}$, we see that:

$$\varphi_{0,j} = \phi_c + \phi_{0,j}$$

Assuming reciprocity in all channels we can apply $\phi_{j,0} = \phi_{0,j}$ $\forall j \in \{1,\ldots,N\}$, from where we deduce that $-\varphi_{0,j}$ is a good choice for the initial phase ϕ_j. For this reason we propose that each user S_j performs an estimate $\hat{\varphi}_{0,j}$ of the initial phase of $y_{0,j}(t)$ and use it as $-\phi_j$. Note that this estimate will be generated from the same data as the frequency estimate.

3.5 Discussion

Unlike the current existing works, this protocol avoids the realization of a high number of frequency and phase estimations. However a considerable number of time-delay estimations is needed, whose errors affect the simultaneity and hence can penalize the resulting beamforming. In order to overcome this limitation, reference signals should allow high accuracy delay estimation, such as Ultra-Wideband signals [9]. Note that, unlike in [8], we have not imposed any constraint on the nature of the reference signals, which leaves freedom to choose those signals that provide better simultaneity at destination, and this facilitates the use of future high data-rate transmissions.

We want to stand out that if $r_0(t)$ and the calibration signal $x_0(t)$ could be sent together, then the calibration procedure could be done in the timeslot TS_0, which would reduce the synchronization time. Note that it is not vital to calibrate before each timing synchronization stage, though.

Regarding to the negative consequences of the sensors disappearances, we remark that the worst cases correspond to situations where deaths occur during the down-cycle, because it would imply that some alive sensors cancel beamforming until the end of the next synchronization stage. Fortunately, the probability that a sensor is alive during the up-cycle and dies during the down-cycle is very low given the short duration of these cycles compared to the beamforming time T_{beam}. For this reason we only need to consider the consequences of sensor disappearances during the up-cycle.

A good measure of the overhead is the ratio R between the wasted time due to deaths, i.e. $(2T_p)m$, and the total synchronization time. Assuming that the total synchronization time when no deaths occur follows an uniform distribution between 0 and $(2N-1)T_p$, the expected value of R for large networks ($N \rightarrow +\infty$) is:

$$\lim_{N\rightarrow+\infty} \mathrm{E}\{R\} = r\log\left(\frac{r+1}{r}\right) \quad r \in (0,1]$$

where r is the death ratio. For $r = 0$, R equals 0. This asymptotic value approximates $\mathrm{E}\{R\}$ very accurately (the error in the approximation is less than 0.01 for $N \geq 15$).

4. SIMULATION RESULTS

This section presents numerical examples of the robust time-slotted round-trip carrier synchronization protocol. We assumed that $x_0(t)$ is sent at a frequency of 900 MHz and has a duration of $T_c = 5\mu$s, where T_c can be large because the base station is not power constrained and the calibration step needs not be carried out very often. We also considered that all signals are received at a SNR of 20 dB.

Monte Carlo simulations with 5000 iterations were executed for all the examples, and at each iteration different estimations of the frequency, phase and time-delay were performed. The errors in the phase, frequency and time-delay estimates were modelled as Gaussian, with zero mean and standard deviation given by the corresponding Cramér-Rao bound. For those cases where we considered sensor disappearances, we selected different nodes to die at each iteration by following an uniform distribution. The error in the correction of τ_j was computed as in (6) and each sensor oscillator deviation factor ε_j was modelled as a Gaussian random variable with unitary mean and standard deviation $\sigma_\varepsilon = 10^{-6}$, which corresponds to a clock with 1 ppm precision.

The example of Fig. 3 shows the probability that the beamforming gain is greater than αN versus the elapsed time since the start of T_{beam}, where $\alpha \in [0,1]$ and N is the maximum achievable gain. All time-delay estimations include errors with standard deviation $\sigma_\tau = 1ps$. In order to show the

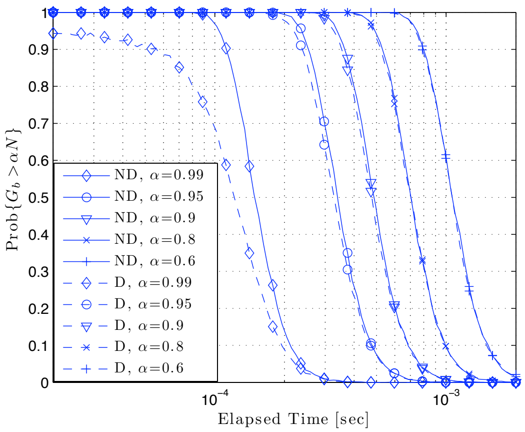

Figure 3: Probability that the beamforming gain (G_b) is greater than a fraction of the number of alive sensors ($N = 10$) versus elapsed time. The label "D" corresponds to deaths with $r = 0.5$. The label "ND" corresponds to no deaths.

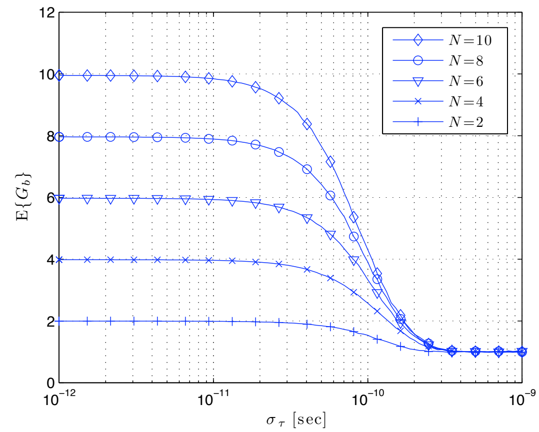

Figure 4: Expected beamforming gain versus delay estimation standard deviation σ_τ for several number of sensors N.

robustness of our protocol we simulated cases where 50% of the sensors disappear during the up-cycle ($r = 0.5$). Higher ratios were considered and the results did not deteriorate up to 70%. We see how the carrier phases are efficiently aligned at the destination up to 600μs, where a 60% quality beamforming no longer can be achieved with high probability.

We want to stand out that simulations of Fig. 3 were done for the same number of alive sensors in order to do a fair comparison. It means that the cases with deaths correspond to an scenario with N alive sensors when initially there were $2N$. On the other hand, the cases with no deaths correspond to an scenario with N sensors and no deaths.

The second example shows the performance of the protocol versus the delay estimation quality σ_τ. Concretely, Fig. 4 plots the achieved beamforming gain after 100μs for different number of sensors. The results for $r = 0$ and $r = 0.5$ were indistinguishable, so we only represent the first case. We can see that errors with $\sigma_\tau < 50ps$ are not deleterious for this protocol. For higher values, the imperfect simultaneity reduces the beamforming gain at the start of the beamforming stage, hence decreasing the beamforming time. In the case that reference signals let $\sigma_\tau = 1ps$, errors can be neglected.

We remark that we also corroborated the behaviour of our algorithm using an event-based implementation of the protocol in MATLAB®, and results were satisfactory.

5. CONCLUSION

In this paper, we have proposed a robust time-slotted round-trip carrier and timing synchronization protocol for distributed beamforming in WSNs. We have described how the protocol prevents the nodes from being blocked and how it avoids cancelling beamforming when some sensors disappear. It has been shown that our proposal is based on a simple exchange of reference signals and a calibration signal.

We have analysed the performance of the protocol and its dependency on time-delay estimation errors. Our numerical results have shown that a good beamforming time can be achieved, even for cases where nodes disappear with death ratios as high as 50%. The effect of the quality of the delay estimations on the final beamforming time has shown that it is necessary to obtain accuracies as good as 50ps. However we leave freedom to choose those signals that provide better simultaneity at destination, which also makes possible the use of future high data-rate systems.

REFERENCES

[1] Y. Hong, W. Huang, F. Chiu, and C. Kuo, "Cooperative Communications in Resource-Constrained Wireless Networks," *IEEE Signal Process. Magazine*, vol. 24, pp. 47–57, May 2007.

[2] H. L. Van Trees, *Optimum Array Processing (Detection, Estimation, and Modulation Theory, Part IV)*, 1st ed. Wiley-Interscience, March 2002.

[3] Y.-S. Tu and G. Pottie, "Coherent cooperative transmission from multiple adjacent antennas to a distant stationary antenna through awgn channels," in *Proc. IEEE Veh. Technol. Conf.*, vol. 1, Birmingham, AL, May 2002, pp. 130–134.

[4] G. Barriac, R. Mudumbai, and U. Madhow, "Distributed beamforming for information transfer in sensor networks," in *Proc. IEEE/ACM Int. Conf. on Inf. Process. in Sens. Net.*, Berkeley, CA, Apr. 2004, pp. 81–88.

[5] R. Mudumbai, J. Hespanha, U. Madhow, and G. Barriac, "Scalable feedback control for distributed beamforming in sensor networks," in *Proc. IEEE Int. Symp. Inf. Theory*, Adelaide, Australia, Sep 2005, pp. 137–141.

[6] R. Mudumbai, G. Barriac, and U. Madhow, "On the feasibility of distributed beamforming in wireless networks," *IEEE Trans. Wireless Commun.*, vol. 6, no. 5, pp. 1754–1763, May 2007.

[7] Q. Wang and K. Ren, "Time-slotted round-trip carrier synchronization in large-scale wireless networks," in *Proc. IEEE Int. Conf. on Commun.*, Beijing, China, May 2008, pp. 5087–5091.

[8] D. Brown and H. Poor, "Time-slotted round-trip carrier synchronization for distributed beamforming," *IEEE Trans. on Signal Process.*, vol. 56, no. 11, pp. 5630–5643, Nov. 2008.

[9] S. Gezici, H. Celebi, H. Arslan, and H. Poor, "Theoretical limits on time delay estimation for ultra-wideband cognitive radios," in *Proc. IEEE Int. Conf. on Ultra-Wideband*, Hannover, Germany, Sep. 2008, pp. 177–180.