Censoring for Bayesian Cooperative Positioning in Dense Wireless Networks

Kallol Das, *Student Member, IEEE*, Henk Wymeersch, *Member, IEEE*

***Abstract*—Cooperative positioning is a promising solution for location-enabled technologies in GPS-challenged environments. However, it suffers from high computational complexity and increased network traffic, compared to traditional positioning approaches. The computational complexity is related to the number of links considered during information fusion. The network traffic is dependent on how often devices share positional information with neighbors. For practical implementation of cooperative positioning, a low-complexity algorithm with reduced packet broadcasts is thus necessary. Our work is built on the insight that for precise positioning, not all the incoming information from neighboring devices is required, or even useful. We show that blocking selected broadcasts (transmit censoring) and discarding selected incoming information (receive censoring) based on a Cramér-Rao bound criterion, leads to an algorithm with reduced complexity and traffic, without significantly affecting accuracy and latency.**

***Index Terms*—Indoor positioning, link selection, cooperative positioning, distributed wireless localization, Cramér Rao bound, censoring.**

I. Introduction

POSITIONAL information is considered to be of great importance in many applications, such as navigation [1], search-and-rescue operations [2], disaster management [3], sensor networks [4], supply chain monitoring [5], and traffic control [6]. Focusing on range-based systems, different techniques are currently available for positioning, which can be classified into two major categories: non-cooperative and cooperative [7]. In a non-cooperative setting, devices rely on distance estimates with reference nodes, whereas in a cooperative setting, devices additionally use distance estimates between each other. These additional measurements can enable positioning in GPS-challenged environments, such as indoors or in urban canyons.

Depending on the use of prior information, cooperative positioning algorithms can be further divided into two categories: non-Bayesian and Bayesian. In non-Bayesian methods, devices exchange position estimates [7], whereas in Bayesian methods, devices exchange full statistical information [8]. While cooperation leads to improved performance, it also results in a high computational complexity per device, due to the additional information from neighboring devices that needs to be processed and fused. Moreover, cooperating devices broadcast their positional information (point estimates or distributions), leading to increased network traffic and packet loss. The impact of packet loss on positioning performance was considered in [9], [10], showing severe degradations. These drawbacks make cooperative positioning algorithms challenging to implement in practice.

When more than the minimum number of reference nodes for positioning is available to a given device, some form of link selection can be applied [11]–[16]. Such link selection can be seen as *information censoring*, previously applied in decentralized detection for sensor networks [17], [18]. For positioning, the use of the closest reference nodes as a censoring criterion was proposed in [11]. The closest reference nodes may not be the most informative for positioning as the geometric configuration also affects the positioning performance. This problem has been partially addressed in [12], [13], where the Cramér-Rao bound (CRB) was used to choose the best reference nodes. In [15], [16] geometric dilution of precision (GDOP) was applied to select the best four satellites for a GPS receiver. None of the methods above are designed for cooperative positioning. Recently, [19] considered non-Bayesian cooperative positioning and proposed to use the neighbors with the highest received signal strength to reduce complexity and energy consumption in sensor nodes. In [20], we have shown that in non-Bayesian cooperative positioning, complexity and traffic can be *reduced simultaneously*, without degrading positioning performance, by using a CRB-based criterion. This is achieved by blocking the broadcasts of the nodes that do not have reliable estimates (transmit censoring) and selecting the most informative links after receiving signals from neighbors (receive censoring).

In this paper, we extend transmit and receive censoring to Bayesian cooperative positioning, where the nodes share full statistical positional information instead of point estimates. Our main contributions are as follows:

- We propose a simple, yet effective censoring criterion based on the modified Bayesian CRB in conjunction with a simple message approximation;
- We show that the complexity of Bayesian cooperative positioning can be reduced significantly, by applying *receive censoring*;
- We show that network traffic can be reduced to some extent when devices block the broadcast of unreliable information, by applying *neighbor-agnostic transmit censoring*;
- We show that the network traffic can be reduced significantly when devices block the broadcast of information

Kallol Das is with the Pervasive Systems Group, University of Twente, Enschede, The Netherlands (Email: k.das@utwente.nl). Henk Wymeersch is with the Department of Signals and Systems, Chalmers University of Technology, Gothenburg, Sweden (Email: henkw@chalmers.se). This research was supported, in part, by the European Research Council, under Grant No. 258418 (COOPNET), and the Swedish Research Council, under Grant No. 2010-5889.

that will be ignored by neighbors, by applying *neighbor-aware transmit censoring*;

- We propose a combined censoring scheme that leads to reduced complexity and reduced traffic, without significantly affecting the positioning performance or the latency.

The remainder of this paper is arranged as follows. In Section II, we describe our model and assumptions. In Section III the censoring criterion is introduced, and then applied to develop three censoring schemes. Results from simulations are discussed in Section IV. Finally, we present our conclusions in Section V.

II. Problem formulation

A. System Model

We consider a wireless network comprising two classes of nodes: agents and anchors. Agents have unknown positions, while anchor have a priori known positions. The goal of the agents is to determine their positions, based on the positions of the anchors and distances estimates between nodes. We denote by $\mathbf{x}_i$ the position of node i and by $S_{\rightarrow i}$ the indices of nodes from which node i can receive signals. Through a ranging protocol (e.g., time of arrival (TOA) or received signal strength (RSS)) with node $j \in \mathcal{S}_{\rightarrow i}$, node i can estimate the distance $\hat{d}_{j\rightarrow i} = \|\mathbf{x}_i - \mathbf{x}_j\| + n_{j\rightarrow i}$, where $n_{j\rightarrow i}$ is the ranging noise. For simplicity, as in [8], we assume $n_{j\rightarrow i} \sim \mathcal{N}\left(0, \sigma^2_{j\rightarrow i}\right)$. Our model assumes all nodes are static, but our findings can be extended to a mobile scenario where nodes move in discrete time slots.

B. Drawbacks of Cooperative Positioning

Different algorithms for cooperative positioning have been proposed (see [7], [8] and references therein). In this paper, we will consider the sum-product algorithm over a wireless network (SPAWN) from [8], as it offers excellent performance with low latency. In SPAWN, every agent has an associated a priori distribution, $b^{(0)}_{\mathbf{X}_i}(\mathbf{x}_i)$. Statistical information is exchanged and computed iteratively through messages, corresponding to distributions of two- or three-dimensional continuous random variables. At every iteration (k), every agent (i) updates its distribution, written as $b^{(k)}_{\mathbf{X}_i}(\mathbf{x}_i)$, named the belief. SPAWN is summarized in Algorithm 1, for a agent i at iteration k. This algorithm is executed in parallel by every agent in the network until the beliefs have converged. Initially, the beliefs $b^{(0)}_{\mathbf{X}_i}(\mathbf{x}_i)$ are set to uniform distributions (which are not broadcast) for the agents and delta Dirac distributions for the anchors. In Algorithm 1, lines 2 and 5 are not part of standard SPAWN, but form the focus of this paper. The messages and beliefs in SPAWN are distributions of multi-dimensional random variables. Exact representation of these distributions is generally impossible, so one must resort to non-parametric [21] or parametric [22] representations. While the representation has a direct impact on the complexity of SPAWN, we will not make any assumptions on the message representation.

As a performance example, we have simulated a 100 meter $\times$100 meter environment with 100 agents having 20 meter communication range and 13 systematically placed anchors [8, see Figure 13], with a ranging noise variance of $\sigma^2_{j\rightarrow i} = (10\,\mathrm{cm})^2$. The positioning performance of SPAWN in terms of the complementary cumulative distribution function (CCDF) of the positioning error at different iterations (from top to bottom) is shown in Figure 1. Observe that after 5 iterations, 99% of the agents have less than 1 meter positioning error. The remaining 1% agents could not satisfactorily converge due to their bad geometrical placement or limited connectivity. Despite the fast convergence and excellent positioning performance, SPAWN suffers from two important drawbacks. First of all, the complexity of SPAWN per agent (to be detailed in Section IV-B1) grows linearly with the number of neighbors. In our example, the average number of links per agent is roughly 13.7, whereas in a non-cooperative environment with the same communication range, this number in only 1.5. Hence, the complexity is almost ten times larger. Secondly, at every iteration of SPAWN, every agent broadcasts a packet, containing its location information. This results in a large amount of network traffic.

Algorithm 1 SPAWN (iteration k, agent i).

1: receive $b^{(k-1)}_{\mathbf{X}_j}(\cdot)$ from neighbors $j \in S_{\rightarrow i}$
2: select the set $S^{(k)}_{\rightarrow i}$ of most informative links through receive censoring
3: convert $b^{(k-1)}_{\mathbf{X}_j}(\cdot)$ to a distribution over $\mathbf{X}_i$

$$m_{\mathbf{X}_j \rightarrow \mathbf{X}_i}(\mathbf{x}_i) \propto \int p(\hat{d}_{j\rightarrow i}|\mathbf{x}_i, \mathbf{x}_j) b^{(k-1)}_{\mathbf{X}_j}(\mathbf{x}_j) \mathrm{d}\mathbf{x}_j$$

4: compute new belief

$$b^{(k)}_{\mathbf{X}_i}(\mathbf{x}_i) \propto b^{(0)}_{\mathbf{X}_i}(\mathbf{x}_i) \prod_{j \in S^{(k)}_{\rightarrow i}} m_{\mathbf{X}_j \rightarrow \mathbf{X}_i}(\mathbf{x}_i)$$

5: decide if transmit censored
6: broadcast $b^{(k)}_{\mathbf{X}_i}(\cdot)$ if not censored

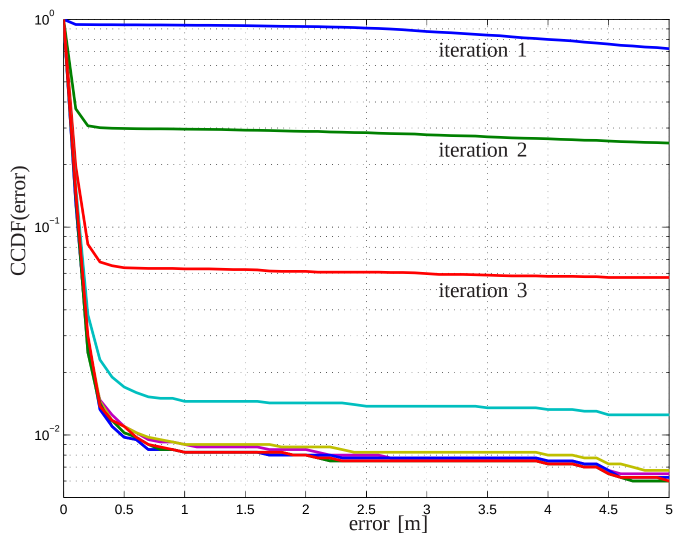

Fig. 1. Positioning performance of SPAWN at different iterations.

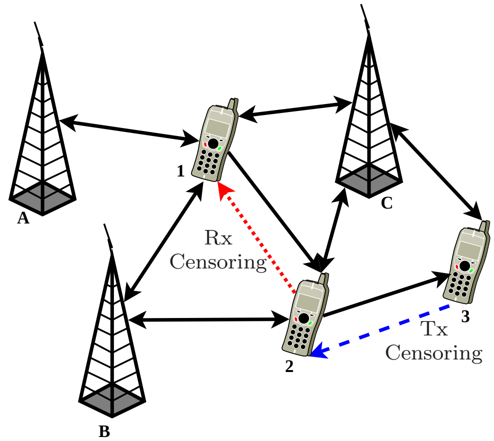

Fig. 2. Transmit and receive censoring schemes in a cooperative network, with 3 agents (1, 2, and 3) and 3 anchors (A, B, and C).

III. Censoring

A. Concept

We will describe the intended censoring schemes by considering a small example with three agents and three anchors, depicted in Figure 2. Agent 3 is connected to only one anchor, so initially it has limited knowledge about its position. Hence, this agent can only provide limited information to its neighbors. In turn, this implies that if this agent blocks the broadcast of its positional information, the overall performance will not be greatly affected. We define this blocking as *transmit censoring* (TxC). Agent 2 is connected to two anchors, which gives it a position ambiguity. Its information may be useful for other agents. So agent 2 should broadcast its positional information. Agent 1 can get information from three anchors and also from agent 2. By ignoring the information from agent 2, its positioning accuracy may be relatively unaffected. We define this ignoring as *receive censoring* (RxC).

B. Censoring Criterion

1) The Modified Bayesian Cramér-Rao Bound: Transmit and receive censoring as intuited in the previous section, require a rigorous criterion based on which agents decide whether or not to censor. This criterion should reflect (i) the quality of the ranging; (ii) the local geometry of the agents and its neighbors; (iii) the uncertainty of the agent's position, and the uncertainty of the neighbors' positions. In addition, the criterion should allow fast computation. One criterion that satisfies these conditions is the modified Bayesian Cramér-Rao bound (MBCRB) [23], defined as follows. Assume that both $\mathbf{x}_i$, the position of the agent in question, and $\{\mathbf{x}_j\}_{j\in S_{\to i}}$, the positions of the neighbors of agent i, are random variables with corresponding distributions[1] $p_{\mathbf{X}_j}(\mathbf{x}_j)$, $j \in S_{\to i}\cup\{i\}$, then the so-called modified Bayesian Fisher information matrix (MBFIM) is defined as

$$\mathbf{F}_i = \underbrace{-\sum_{j\in S_{\to i}} \mathbb{E}_{n_{j\to i},\mathbf{x}_i,\mathbf{x}_j}\left\{\frac{\partial^2 \log p\left(\hat{d}_{j\to i}\mid \mathbf{x}_i,\mathbf{x}_j\right)}{\partial \mathbf{x}_i^2}\right\}}_{=\mathbf{F}_{\mathrm{M},i}} \underbrace{-\mathbb{E}_{\mathbf{x}_i}\left\{\frac{\partial^2 \log p_{\mathbf{X}_i}(\mathbf{x}_i)}{\partial \mathbf{x}_i^2}\right\}}_{=\mathbf{F}_{\mathrm{P},i}}. \quad (1)$$

The expectation in (1) occurs over the ranging noise and the nodes' positions. The MBFIM can be broken up into a term related to measurements ($\mathbf{F}_{\mathrm{M},i}$) and a term related to a priori information ($\mathbf{F}_{\mathrm{P},i}$). When $p_{\mathbf{X}_i}(\mathbf{x}_i)$ is a Gaussian distribution with covariance matrix $\mathbf{\Sigma}_i$, then $\mathbf{F}_{\mathrm{P},i} = \mathbf{\Sigma}_i^{-1}$. Assuming Gaussian ranging noise, the expectation over the ranging noise can be carried out analytically [24], leading to

$$\mathbf{F}_{\mathrm{M},i} = \sum_{j\in S_{\to i}} \frac{1}{\sigma^2_{j\to i}} \mathbb{E}_{\mathbf{x}_i,\mathbf{x}_j}\left\{\frac{\mathbf{x}_i-\mathbf{x}_j}{\|\mathbf{x}_i-\mathbf{x}_j\|}\frac{(\mathbf{x}_i-\mathbf{x}_j)^{\mathrm{T}}}{\|\mathbf{x}_i-\mathbf{x}_j\|}\right\}. \quad (2)$$

The expectation over $\mathbf{x}_i$ and $\mathbf{x}_j$ is generally difficult to perform analytically, so we resort to Monte Carlo integration. Assuming we can draw N samples $\{\mathbf{x}_j^{(n)}\}_{n=1}^N$ from $p_{\mathbf{X}_j}(\cdot)$, $j\in S_{\to i}\cup\{i\}$, we find that

$$\begin{aligned}\mathbf{F}_{\mathrm{M},i} &= \sum_{j\in S_{\to i}} \frac{1}{\sigma^2_{j\to i}} \int p_{\mathbf{X}_i}(\mathbf{x}_i)p_{\mathbf{X}_j}(\mathbf{x}_j)\frac{\mathbf{x}_i-\mathbf{x}_j}{\|\mathbf{x}_i-\mathbf{x}_j\|}\frac{(\mathbf{x}_i-\mathbf{x}_j)^{\mathrm{T}}}{\|\mathbf{x}_i-\mathbf{x}_j\|}\mathrm{d}\mathbf{x}_i\mathrm{d}\mathbf{x}_j \\ &\approx \frac{1}{N}\sum_{j\in S_{\to i}} \frac{1}{\sigma^2_{j\to i}} \sum_{n=0}^{N-1} \frac{\mathbf{x}_i^{(n)}-\mathbf{x}_j^{(n)}}{\left\|\mathbf{x}_i^{(n)}-\mathbf{x}_j^{(n)}\right\|}\frac{\left(\mathbf{x}_i^{(n)}-\mathbf{x}_j^{(n)}\right)^{\mathrm{T}}}{\left\|\mathbf{x}_i^{(n)}-\mathbf{x}_j^{(n)}\right\|}. \quad (3)\end{aligned}$$

Finally, the MBCRB can be calculated as

$$\mathrm{MBCRB}_i = \mathrm{trace}\left(\mathbf{F}_i^{-1}\right). \quad (4)$$

This MBCRB_i is also defined when $S_{\to i} = \emptyset$, i.e., even when there are no neighbors for the update, or when there are no measurements. In this case, $\mathrm{MBCRB}_i = \mathrm{trace}\left(\mathbf{\Sigma}_i\right)$. Incidentally, we note that when $p_{\mathbf{X}_i}(\mathbf{x}_i)$ is uniform, and $p_{\mathbf{X}_j}(\mathbf{x}_j)$, $j\in S_{\to i}$ are delta Dirac distributions, (4) reverts to the censoring criterion considered in [20].

2) Message Approximation for Censoring: In order to be able to compute the MBCRB efficiently, the distributions $p_{\mathbf{X}_j}(\mathbf{x}_j)$ should not be too complex. On the other hand, the true beliefs $b^{(k)}_{\mathbf{X}_i}(\mathbf{x}_i)$ can have many different shapes. For our purpose, the details of the shape of $b^{(k)}_{\mathbf{X}_i}(\mathbf{x}_i)$ are not so important, but rather we wish to capture how concentrated the distribution is, and the position of the centers of mass. A simple Gaussian approximation is not sufficient as it cannot capture the common case when an agent can communicate with two anchors, leading to a bimodal belief with two highly concentrated components. Hence, we propose to approximate the beliefs with a mixture of two Gaussians: we first determine the number of components $N_i^{(k)} \in \{1,2\}$ of the belief $b^{(k)}_{\mathbf{X}_i}(\mathbf{x}_i)$ of agent $\mathbf{x}_i$ at iteration k. For every component, we then determine the mean ($\boldsymbol{\mu}^{(k)}_{1,i}$ and $\boldsymbol{\mu}^{(k)}_{2,i}$) and the covariance

[1] As we will see later, these distributions will be simple approximations to the beliefs $b^{(k)}_{\mathbf{X}_j}(\mathbf{x}_j)$, computed in SPAWN.

Algorithm 2 Receive censoring for SPAWN (iteration k, agent i).

1: receive $b^{(k-1)}_{\mathbf{X}_j}(\mathbf{x}_j)$ from neighbors, $j \in S_{\to i}$
2: **if** $N_i^{(k-1)} = 1$ **then**
3: **if** trace $\left(\boldsymbol{\Sigma}_i^{(k-1)}\right) < \gamma_{\mathrm{RX}}$ **then**
4: set $S^{(k)}_{\to i} = \emptyset$
5: **else**
6: select L neighbors from $S_{\to i}$: see Algorithm 3
7: **end if**
8: **else**
9: remove ambiguity in $b^{(k-1)}_{\mathbf{X}_i}(\mathbf{x}_i)$
10: **goto** line 3
11: **end if**
12: use $S^{(k)}_{\to i}$ for update

matrix ($\boldsymbol{\Sigma}^{(k)}_{1,i}$ and $\boldsymbol{\Sigma}^{(k)}_{2,i}$). For simplicity and robustness, we further only consider the covariance matrix with the largest trace: $\boldsymbol{\Sigma}^{(k)}_i = \arg\max_{\boldsymbol{\Sigma} \in \left\{\boldsymbol{\Sigma}^{(k)}_{1,i}, \boldsymbol{\Sigma}^{(k)}_{2,i}\right\}} \mathrm{trace}\,(\boldsymbol{\Sigma})$. Finally, we approximate all beliefs at iteration k by a mixture of two Gaussians as $p^{(k)}_{\mathbf{X}_i}(\mathbf{x}_i) \approx b^{(k)}_{\mathbf{X}_i}(\mathbf{x}_i)$, where

$$p^{(k)}_{\mathbf{X}_i}(\mathbf{x}_i) = \frac{1}{2}\mathcal{N}\left(\boldsymbol{\mu}^{(k)}_{1,i}, \boldsymbol{\Sigma}^{(k)}_i\right) + \frac{1}{2}\mathcal{N}\left(\boldsymbol{\mu}^{(k)}_{2,i}, \boldsymbol{\Sigma}^{(k)}_i\right). \quad (5)$$

When $N_i^{(k)} = 1$, we have that $\boldsymbol{\mu}^{(k)}_{1,i} = \boldsymbol{\mu}^{(k)}_{2,i}$. We note that this approximation is used only within the censoring methods, while the messages computed and propagated in SPAWN remain unaffected. More sophisticated approximations to $b^{(k)}_{\mathbf{X}_i}(\mathbf{x}_i)$ can of course be considered, but as we will see, a mixture of two Gaussians is sufficient for our scenario.

C. Censoring Schemes

1) Neighbor-Agnostic Transmit Censoring: In neighbor-agnostic transmit censoring, an agent will decide to broadcast or censor its positional information based on the uncertainty of its own belief. After calculating its belief $b^{(k)}_{\mathbf{X}_i}(\mathbf{x}_i)$ at iteration k, agent i can determine the covariance matrix $\boldsymbol{\Sigma}^{(k)}_i$ associated with $p^{(k)}_{\mathbf{X}_i}(\mathbf{x}_i)$, indicating how concentrated the belief is. An agent will transmit-censor when

$$\mathrm{trace}\left(\boldsymbol{\Sigma}^{(k)}_i\right) \geq \gamma_{\mathrm{TX}}. \quad (6)$$

The transmit censoring threshold γ_{TX}, expressed in m^2, depends on the ranging model and the performance requirements. During the first iteration (non-cooperative phase) agents that can only communicate with zero or one anchors will have beliefs that are not concentrated. Hence, these agents will censor their beliefs. In later iterations, agents can obtain more information from neighbors, leading to more concentrated beliefs, and thus less transmit censoring. We note that the censoring criterion does not directly depend on the neighbors' beliefs. For that reason, we call this censoring scheme neighbor-agnostic.

Algorithm 3 Link selection of L most informative links.

1: **if** $|S_{\to i}| > L$ **then**
2: create $\mathcal{S}_L = \{\text{all subsets of } S_{\to i} \text{ of size } L\}$
3: **for** $l = 1$ to $|\mathcal{S}_L|$ **do** {subset index}
4: let $\mathcal{S}_L[l]$ be the l-th subset in $\mathcal{S}_L$
5: determine

$$\mathrm{MBCRB}_i[l] = \mathrm{trace}\left(\mathbf{F}_i^{-1}[l]\right),$$

where

$$\mathbf{F}_i[l] = \sum_{j \in \mathcal{S}_L[l]} \frac{1}{\sigma^2_{j \to i}} \mathbb{E}_{\mathbf{x}_i, \mathbf{x}_j}\left\{\frac{\mathbf{x}_i - \mathbf{x}_j}{\|\mathbf{x}_i - \mathbf{x}_j\|} \frac{(\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}}}{\|\mathbf{x}_i - \mathbf{x}_j\|}\right\} + \left[\boldsymbol{\Sigma}^{(k-1)}_i\right]^{-1}$$

6: **end for**
7: select the best subset

$$\hat{l} = \arg\min_{l} \mathrm{MBCRB}_i[l]$$

8: set $S^{(k)}_{\to i}$ to $\mathcal{S}_L[\hat{l}]$
9: **else**
10: set $S^{(k)}_{\to i}$ to $S_{\to i}$
11: **end if**

2) Receive Censoring: In receive censoring, an agent will decide to discard uninformative incoming information from neighboring agents. To allow prior Fisher information of the form $\mathbf{F}_{\mathrm{P},i} = \boldsymbol{\Sigma}_i^{-1}$, we perform a separate pre-processing step to remove ambiguities (see Algorithm 2): based on its belief $b^{(k-1)}_{\mathbf{X}_i}(\mathbf{x}_i)$ at the previous iteration, agent i can determine the covariance matrix $\boldsymbol{\Sigma}^{(k-1)}_i$ and the number of components $N_i^{(k-1)} \in \{1, 2\}$. When $N_i^{(k-1)} = 2$, the agent will try to remove the ambiguity in its belief by considering the information from the neighbors. Ambiguity removal can simply be performed by checking the consistency between the components in $b^{(k-1)}_{\mathbf{X}_i}(\mathbf{x}_i)$ and the beliefs of all the neighbors $b^{(k-1)}_{\mathbf{X}_j}(\mathbf{x}_j)$, $j \in S_{\to i}$. After ambiguity removal,[2] a link selection algorithm (see Algorithm 3) is executed to select the most informative subset of $L \geq 3$ neighbors. However, when trace $\left(\boldsymbol{\Sigma}^{(k-1)}_i\right) < \gamma_{\mathrm{RX}}$, the agent discards all incoming information[3] by setting $S^{(k)}_{\to i} = \emptyset$. The size of the subset (indicated by L in Algorithm 3) should be at least 3 for two-dimensional positioning.

3) Neighbor-Aware Transmit Censoring: While transmit censoring as described in Section III-C1 can reduce the network traffic, it is clear that in combination with receive censoring further reductions in network traffic are possible: when *all* neighbors of agent i satisfy the receive censoring

[2]When the ambiguity in $b^{(k-1)}_{\mathbf{X}_i}(\mathbf{x}_i)$ cannot be removed, line 6 of Algorithm 2 is executed based on one arbitrarily chosen component of $b^{(k-1)}_{\mathbf{X}_j}(\mathbf{x}_i)$. This approximation turns out to have little impact on the final performance, as this case only occurs when agent i or all of its neighbors have beliefs that are not concentrated.

[3]Essentially, we consider the agent as well-localized, so no further processing is required.

Algorithm 4 Neighbor-aware transmit censoring for SPAWN (iteration k, agent i).

```
1: if trace(Σ_i^(k)) > γ_TX then
2:    block the broadcast of b_{X_i}^(k)(x_i)
3: else
4:    broadcast = FALSE
5:    for j = 1 to |S_{→i}| do {neighbor's index}
6:       if N_j^(k−1) = 2 OR trace(Σ_j^(k−1)) > γ_RX then
7:          broadcast = TRUE
8:          break
9:       end if
10:   end for
11:   if broadcast then
12:      broadcast b_{X_i}^(k)(x_i)
13:   end if
14: end if
```

threshold,[4] the broadcasts of agent i will be ignored by all neighbors. Hence, those broadcasts are unnecessary. We can thus develop a neighbor-aware transmit censoring scheme, as outlined in Algorithm 4, which blocks broadcasts that will be ignored by all the neighbors [25]. Observe that neighbor-agnostic transmit censoring corresponds to lines 1–3 in Algorithm 4. It is important to note that this scheme suffers from a hidden node problem: when an agent is not aware a neighbor is present (due to packet loss, transmit censoring, or asymmetric links), it may decide to transmit censor too early.

IV. Numerical Results

A. Simulation Setup

We consider random networks similar to those described in Section II-B, with 100 randomly placed agents, 13 anchors, a 100 m by 100 m map, 20 m communication radius, and 10 cm ranging noise standard deviation. Our focus is on a line-of-sight (LOS) scenario, though the censoring methods can be applied unaltered in non-LOS (NLOS) conditions when NLOS detection is employed [26].

We will first fix the receive censoring threshold γ_{RX} and transmit censoring threshold γ_{TX}, both expressed in m^2. The value of γ_{RX} is directly related to desired positioning accuracy, with more aggressive censoring (i.e., larger values of γ_{RX}) leading to faster convergence, lower complexity, but a reduction in accuracy. Receive censoring is switched off when $\gamma_{\mathrm{RX}} = 0$. The value of γ_{TX} reflects when an agent is deemed informative for neighbors. More aggressive censoring (i.e., smaller value of γ_{TX}) leads to fewer broadcasts, as only highly informative information is shared, but also to less information in the network. Transmit censoring is switched off when $\gamma_{\mathrm{TX}} = +\infty$. For combined transmit and receive censoring, we require that $\gamma_{\mathrm{TX}} \geq \gamma_{\mathrm{RX}}$: when an agent's belief has met the receive censoring threshold, it should not block its broadcasts. We have chosen $\gamma_{\mathrm{RX}} = (0.28\,\mathrm{m})^2$ and $\gamma_{\mathrm{TX}} = (0.45\,\mathrm{m})^2$, which are both on the order of the ranging noise variance. As we will see in Section IV-B5, the system is not very sensitive to the value of either threshold. We set $L = 3$ in Algorithm 3.

We will denote by *NoC* the SPAWN algorithm with no censoring, by *TxC* when only neighbor-agnostic transmit censoring is used, by *RxC* when only receive censoring is used, and by *TxRxC* when receive censoring with neighbor-aware transmit censoring is used.

B. Simulation Discussion

1) Reduction in Complexity: The complexity of SPAWN is mainly related to the number of messages used during message multiplication (line 4 in Algorithm 1). In particular, in a sample-based message representation, the complexity of the message multiplication scales as $\mathcal{O}(Q^2 \left|S_{\to i}^{(k)}\right|)$, where Q is the number of samples per message (typically 1000 – 10000) and $\left|S_{\to i}^{(k)}\right|$ denotes the cardinality of the set $S_{\to i}^{(k)}$. In a parametric message representation, the complexity scales as $\mathcal{O}(C \left|S_{\to i}^{(k)}\right|)$, where C is a (generally large) constant related to the computation of the message parameters, which typically involves solving a non-convex optimization problem [22]. As we will see later, without link selection, $\left|S_{\to i}^{(k)}\right| \approx 10$, while with link selection $\left|S_{\to i}^{(k)}\right| \leq 3$. The complexity of the link selection algorithm (Algorithm 3) scales as $\mathcal{O}\left(N\binom{\left|S_{\to i}^{(k)}\right|}{L}\right)$, where N from (3) is relatively small (say, 200). For small L, the link selection process is much less complex than the message multiplication, which directly motivates the need to reduce the number of multiplied messages. The complexity can be further reduced by performing a greedy, rather than exhaustive search of the L most informative links in Algorithm 3. As a indication, Table I shows the normalized simulation times for a parametric message representation [27]. We observe that with the TxRxC strategy, SPAWN can be executed roughly 19 times faster than without censoring. For a non-parametric representation, results (not shown) indicated similar complexity reductions.

Figure 3 shows, as a function of the iteration index, the average number of multiplied messages per agent for the different censoring strategies. When no censoring is employed, over 10 links are considered per agent at every iteration (except the first, non-cooperative iteration). The TxC strategy results in a marginal reduction, as some broadcasts are blocked. In contrast, the RxC strategy leads to a significant reduction in the number of links used. After two iterations, most agents meet the receive censoring threshold, so the number of links will be close to zero. The combination TxRxC leads to additional

TABLE I
Normalized simulation time for SPAWN with different censoring schemes, for 10 iterations.

	SPAWN NoC	TxC	RxC	TxRxC
simulation time [normalized]	18.9	18.2	1.1	1.0

[4] I.e., $\mathrm{trace}(\mathbf{\Sigma}_j^{(k-1)}) < \gamma_{\mathrm{RX}}$ AND $N_j^{(k-1)} = 1$, $\forall j \in S_{\to i}$.

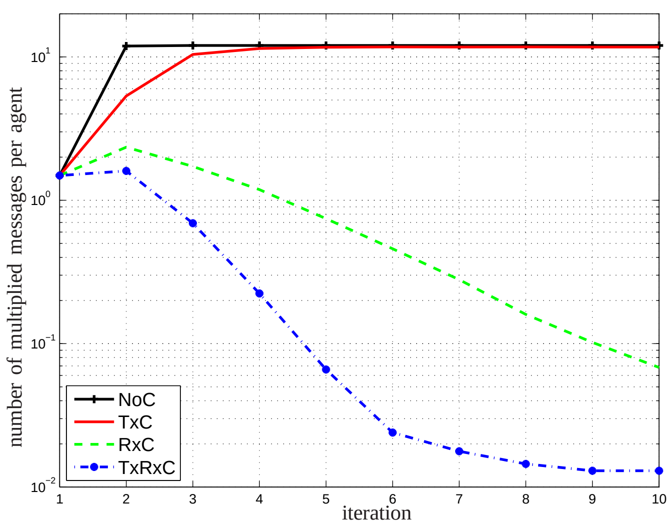

Fig. 3. Complexity: comparison of the average number of used links (or messages multiplied) for different censoring schemes.

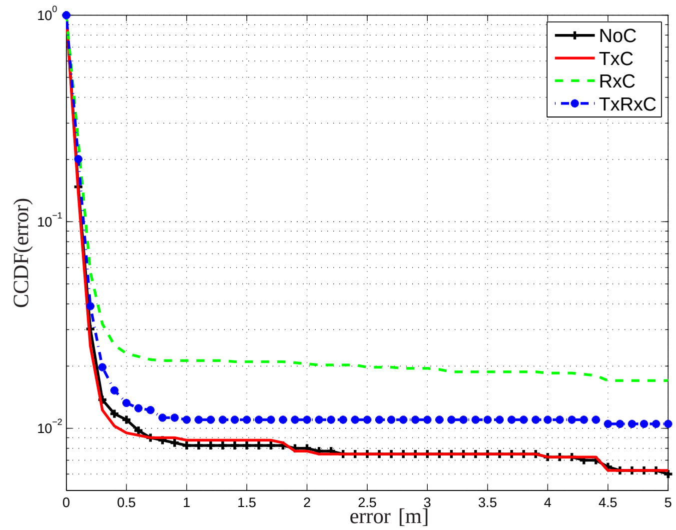

Fig. 5. Positioning performance comparison after 10 iterations with and without censoring.

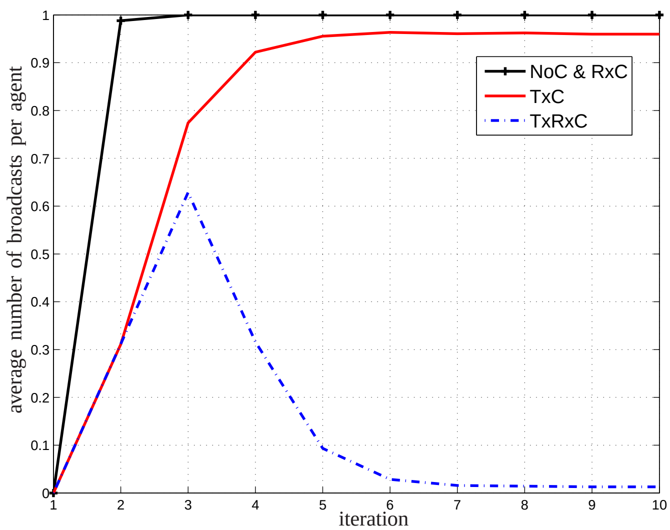

Fig. 4. Network traffic: comparison of average number of broadcasts per agent, for different censoring schemes.

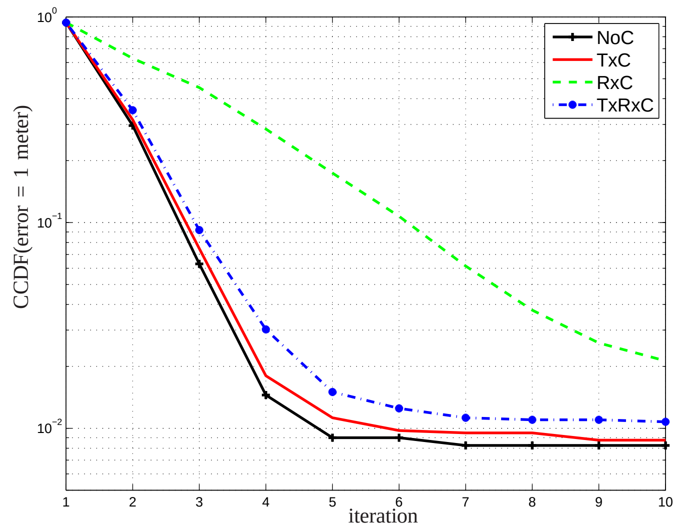

Fig. 6. Convergence speed of different censoring schemes.

gains, as agents are more likely to receive useful information, and hence more quickly meet the receive censoring threshold. The quantitative reduction in complexity due to the reduction in the number of multiplied messages depends on the particular message representation.

2) Reduction in Network Traffic : The high network traffic in SPAWN is due to every agent broadcasting its belief at every iteration. Figure 4 shows, as a function of the iteration index, the average number of broadcasts per agent for the different censoring strategies. Without censoring, almost every agent will broadcast its belief at every iteration, except the first one. Applying the TxC strategy results in a reduction of the number of broadcasts, especially in the first few iterations, when many agents are not yet well-localized. In later iterations, when most agents are well-localized, no censoring takes place, resulting in almost the same number of broadcasts compared to conventional SPAWN. The TxRxC strategy follows the same trend as TxC for the first few iterations. Then, neighbor-aware transmit censoring can harness the fact that an agent's neighbors have met the receive censoring threshold and block that agent's broadcast. Hence, the average number of broadcasts drop close to zero with further iterations.

3) Positioning Performance : We now investigate the positioning performance of the different censoring schemes. Figure 5 shows complementary cumulative distribution function (CCDF) of the positioning error, i.e., the probability that the positioning error exceeds a certain value, after 10 iterations. We can observe that the CCDF of TxC follows the CCDF of conventional SPAWN because most of the available links are used (see also Figure 3). On the other hand, RxC results in a performance degradation compared to conventional SPAWN, as agents only use a subset of $L = 3$ links from the available links during message multiplication. Interestingly, the TxRxC strategy outperforms RxC. The reason for this is that uninformative beliefs are transmit-censored, so that during receive censoring, the links to choose from all correspond to concentrated beliefs.

4) Convergence Speed: The convergence speed of the algorithm is directly related to the latency and the refresh-rate. In Figure 6 we compare the positioning performance as a function of the iteration index. The positioning performance is measured in terms of the CCDF at a fixed value of the positioning error (1 m). For instance, a curve with label "TxC" shows, under the TxC strategy, the probability that an agent will have a positioning error greater than 1 m, at every iteration. We observe that RxC converges the slowest, while TxC and TxRxC require 5–6 iteration to converge, irrespective of the error value.

5) Sensitivity to Parameters γ_{TX}, γ_{RX}, and L: We varied γ_{TX}, γ_{RX}, and L. Changing γ_{TX} around $(0.45\,\mathrm{m})^2$ did not lead to a significant change in performance or traffic, but too conservative transmit censoring causes increased network traffic. Any change in γ_{RX} affects the complexity of the algorithm through the average number of used links, as well as the required number of iterations for convergence. By reducing γ_{RX} to $(0.14\,\mathrm{m})^2$, the gap in positioning performance between NoC and RxC can be reduced significantly, at a small cost in complexity, as fewer agents meet the receive censoring threshold. Finally, changing L from 3 to 4 did not yield any significant performance improvement, but results in additional complexity in Algorithm 3.

V. Conclusions and Future Work

Motivated by the need to reduce complexity and network traffic in cooperative positioning schemes, we have proposed and evaluated different censoring schemes. All censoring decisions are distributed and based on a modified Bayesian Cramér-Rao bound criterion. By applying the proposed censoring schemes to Bayesian cooperative positioning, we have found that: (i) receive censoring (ignoring uninformative information) can dramatically reduce the complexity of information fusion, but at the cost in positioning performance; (ii) transmit censoring (blocking broadcasts of unreliable information) can reduce the network traffic during first few iterations without positioning performance degradation; (iii) receive censoring with neighbor-aware transmit censoring (blocking broadcasts of information that will be ignored) can further significantly reduce the network traffic. Overall, this latter scheme maintains the excellent performance and low latency of Bayesian cooperative positioning without censoring, but does so at a fraction of the computational cost, and at a fraction of the network traffic. These advantages of censoring schemes, along with their distributed nature make them promising for large-scale dense networks. Future work includes extending the proposed censoring schemes to account for NLOS conditions without explicit NLOS detection, as well as a testbed implementation.

References

[1] E. Kaplan, Ed., *Understanding GPS: Principles and Applications*. Artech House, 1996.

[2] J. Jennings, G. Whelan, and W. Evans, "Cooperative search and rescue with a team of mobile robots," in *Proceedings of the 8th International Conference on Advanced Robotics (ICAR)*, 1997, pp. 193–200.

[3] U. Walder, T. Bernoulli, and T. Wießflecker, "An indoor positioning system for improved action force command and disaster management," in *Proceedings of the 6th International ISCRAM Conference*, May 2009.

[4] S. Gezici, Z. Tian, G. Giannakis, H. Kobayashi, A. Molisch, H. Poor, and Z. Sahinoglu, "Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 70–84, 2005.

[5] R. Angeles, "RFID technologies: supply-chain applications and implementation issues," *Information Systems Management*, vol. 22, no. 1, pp. 51–65, 2005.

[6] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 425–437, 2002.

[7] N. Patwari, J. Ash, S. Kyperountas, A. Hero III, R. Moses, and N. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54–69, 2005.

[8] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 427 –450, 2009.

[9] C. Mensing and J. J. Nielsen, "Centralized cooperative positioning and tracking with realistic communications constraints," in *Proceedings of the 7th Workshop on Positioning Navigation and Communication (WPNC)*, 2010, pp. 215–223.

[10] S. Severi, G. Abreu, G. Destino, and D. Dardari, "Efficient and accurate localization in multihop networks," in *2009 Conference Record of the 43rd Asilomar Conference on Signals, Systems and Computers*, 2009, pp. 1071–1076.

[11] V. Tam, K. Cheng, and K. Lui, "Using micro-genetic algorithms to improve localization in wireless sensor networks," *Journal of Communications*, vol. 1, no. 4, pp. 1–10, 2006.

[12] D. Lieckfeldt, J. You, and D. Timmermann, "Distributed selection of references for localization in wireless sensor networks," in *Proceedings of the 5th Workshop on Positioning, Navigation and Communication (WPNC)*, 2008, pp. 31–36.

[13] ——, "An algorithm for distributed beacon selection," in *Proceedings of the 6th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2008, pp. 318–323.

[14] K.-Y. Cheng, V. Tam, and K.-S. Lui, "Improving APS with anchor selection in anisotropic sensor networks," in *Proceedings of Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services, (ICAS-ICNS)*, 2005, pp. 49–54.

[15] J. Li, A. Ndili, L. Ward, and S. Buchman, "GPS receiver satellite/antenna selection algorithm for the Stanford gravity probe B relativity mission," in *National Technical Meeting'Vision 2010: Present and Future*. Institute of Navigation, San Diego, CA, 1999, pp. 541–550.

[16] C. Park, "Precise relative navigation using augmented CDGPS," Ph.D. dissertation, Stanford University, Jun. 2001.

[17] W.-P. Tay, J. N. Tsitsiklis, and M. Z. Win, "Asymptotic performance of a censoring sensor network," *IEEE Transactions on Information Theory*, vol. 53, no. 11, pp. 4191–4209, 2007.

[18] C. Rago, P. Willett, and Y. Bar-Shalom, "Censoring sensors: a low-communication-rate scheme for distributed detection," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 2, pp. 554–568, 1996.

[19] A. Bel, J. Vicario, and G. Seco-Granados, "Real-time path loss and node selection for cooperative localization in wireless sensor networks," in *Proceedings of 2010 IEEE 21st International Symposium on Personal, Indoor and Mobile Radio Communications Workshops (PIMRC Workshops)*, 2010, pp. 283 –288.

[20] K. Das and H. Wymeersch, "Censored cooperative positioning for dense wireless networks," in *Proceedings of 2010 IEEE 21st International Symposium on Personal, Indoor and Mobile Radio Communications Workshops (PIMRC Workshops)*, 2010, pp. 262–266.

[21] A. T. Ihler, J. W. Fisher III, R. L. Moses, and A. S. Willsky, "Nonparametric belief propagation for self-localization of sensor networks," *IEEE Journal on Selected Topics in Communications*, vol. 23, no. 4, pp. 809–819, Apr. 2005.

[22] M. Caceres, F. Penna, H. Wymeersch, and R. Garello, "Hybrid GNSS-terrestrial cooperative positioning via distributed belief propagation," in *Proceeding of IEEE Global Telecommunications Conference*, 2010, pp. 1–5.

[23] H. Van Trees, K. Bell, and S. Dosso, Eds., *Bayesian Bounds for Parameter Estimation and Nonlinear Filtering/Tracking*. John Wiley & Sons, Inc., 2007.

[24] N. Patwari, A. Hero III, M. Perkins, N. Correal, and R. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Transaction on Signal Processing*, vol. 51, no. 8, pp. 2137–2148, 2003.

[25] K. Das and H. Wymeersch, "A network traffic reduction method for cooperative positioning," in *Proceedings of the 8th Workshop on Positioning Navigation and Communication (WPNC)*, 2011.

[26] S. Maranò, W. Gifford, H. Wymeersch, and M. Z. Win, "Nonparametric obstruction detection for UWB localization," in *Proceeding of IEEE Global Telecommunications Conference*, Honolulu, HI, Nov.-Dec. 2009.

[27] W. Li, "Message representation and updates for cooperative positioning," Master's thesis, Department of Signals and Systems, Chalmers University of Technology, 2010.