A Networked Transferable Belief Model approach for Distributed Data Aggregation - Static Version

Flavio Fiorini, Andrea Gasparri, Maurizio Di Rocco and Stefano Panzieri.

Abstract—In this paper the data aggregation problem for a multi-agent system is investigated. In this framework, agents are assumed to be independent reliable sources which collect data and collaborate to reach a common knowledge. In particular, each agent is supposed to provide an observation which does not change over time, i.e., static scenario. A protocol for distributed data aggregation which is proved to converge to the basic belief assignment (BBA) given by a centralized aggregation based on the Transferable Belief Model (TBM) is provided.

I. INTRODUCTION

Data fusion is a research area that is growing rapidly due to the fact that it provides means for combining pieces of information coming from different sources/sensors. As a result, an enhanced overall system performance, i.e., improved decision making, increased detection capabilities, diminished number of false alarms, improved reliability, with respect to separate sensors/sources can be achieved [1]. Indeed, data fusion techniques play an important role in the context of multi-agent systems where information coming from different sources must be aggregated in order to provide a meaningful description of the surrounding environment. The majority of works available in the literature are based on the Bayesian framework, where the aggregation is achieved by applying the Bayes rule. The most representative example is the Kalman Filter, where noisy data is assumed to be described by means of a Gaussian probability distribution [2]. Several works have been proposed to deal with the multi-agent data fusion in a Bayesian framework [3], [4]. In this context, the network topology of the multi-agent system is considered as a Bayesian network over which a message passing algorithm for performing inference on a graphical model is devised.

The Theory of Evidence introduced by Arthur P. Dempster and Glenn Shafer (DS) represents a valid alternative to the Bayesian framework [5]. The main difference concerns the way in which the uncertainty is handled: in the probabilistic framework the uncertainty is treated by splitting the amount of credibility among plausible events, in the DS framework a belief is assigned to the set describing all the plausible hypotheses without supporting any in particular. Depending on the specific application, one framework can be more adequate than the other [6]. This framework was further extended by the Transferable Belief Model (TBM) introduced by Philippe Smets [7]. In particular, TBM introduces the idea of open word assumption in the DS framework. This implies the set of hypotheses not to be exhaustive, therefore data can take to contradiction. Indeed, the concept of contradiction is a powerful tool to detect cases where information fusion has to be considered unreliable, case that is not considered in the Bayesian technique. The main limitation of this framework is the computational complexity, which grows exponentially with respect to the number of events, as their supersets have to be taken into account. To overcome this drawback, several approximation technique have been proposed [8], [9]. However, in case a minimal number of events is enough to model the problem, the TBM approach has been effectively used, e.g. in diagnostic applications [10] and target identification [11].

In this paper the data aggregation problem for a multi-agent system is investigated. In this framework, agents are assumed to be independent reliable sources which collect data and collaborate to reach a common knowledge. A protocol for distributed data aggregation which is proved to converge to the basic belief assignment (BBA) given by a centralized aggregation based on the Transferable Belief Model (TBM) conjunctive rule is provided. The main contribution of this work is the development and the theoretical characterization of a distributed interaction rule to aggregate data within the TBM framework.

The rest of the paper is organized as follows. In Section II some basic concepts concerning the Theory of Evidence are reviewed. In section III an introductory example to explain the problems related to the use of the TBM in a distributed context is given. In Section IV the problem setting along with basic assumptions for the proposed framework is described. In Section V the proposed distributed data aggregation protocol and its convergence properties are described. And finally, in Section VI conclusions are drawn and future work is discussed.

II. THEORY OF EVIDENCE

The Theory of Evidence is a formalism which can be used for modeling uncertainty instead of classical probability. Theory of Evidence embraces the familiar idea of using a number between zero and one to indicate the degree of belief for a proposition on the basis of the available evidence.

Let \( \Omega = \{\omega_1, \ldots, \omega_n\} \) be a finite set of possible values of a variable \( \omega \), where the elements \( \omega_i \) are assumed to be mutually exclusive and exhaustive. Let \( \Gamma(\Omega) \equiv 2^\Omega = \{\gamma_1, \ldots, \gamma|\Gamma|\} \) be the power set associated to it. In this framework, the interest is focused in quantifying the belief of propositions of the form: “the true value of \( \omega \) is in \( \gamma \), with \( \gamma \in \Gamma \). The propositions of interest are
therefore in one-to-one correspondence with the subset $\Omega$, and the set of all propositions of interest corresponds to the elements of $\Gamma$. The set $\Omega$ so defined, is referred to as frame of discernments.

**Definition 1 (BBA):** A function $m : 2^\Omega \rightarrow [0, 1]$ is called a basic belief assignment if

$$m(\emptyset) = 0 \quad (1)$$

$$\sum_{\gamma_a \in \Gamma} m(\gamma_a) = 1 \quad (2)$$

Thus for $\gamma_a \in \Gamma$, $m(\gamma_a)$ is the part of belief that supports exactly $\gamma_a$, i.e. the fact that the true value of $\omega$ is in $\gamma_a$, but due to the lack of further information, does not support any strict subset of $\gamma_a$. The first condition reflects the fact that no belief should be committed to $\emptyset$ and the second condition reflects that the total belief has measure one.

Notice that $m(\gamma_a)$ and $m(\gamma_b)$ can be both equal to zero even if $m(\gamma_a \cup \gamma_b) \neq 0$. Further, $m(\cdot)$ is not monotone under inclusion, i.e. $\gamma_a \subset \gamma_b$ does not imply $m(\gamma_a) < m(\gamma_b)$.

Notice that the BBA represents the atomic information in the theory of evidence.

**Definition 2:** A function $Bel : 2^\Omega \rightarrow [0, 1]$ is called belief function over $\Omega$ if it satisfies the following relationship:

$$Bel(\gamma_a) = \sum_{\gamma_b \subseteq \gamma_a} m(\gamma_b) \quad (3)$$

this function quantifies the total specific amount of belief supporting the event, and it is often taken into account in the decision making process after data aggregation has been performed [7].

The main criticism to Shafer formulation concerns the application of the Dempster-Shafer (DS) combination rule. In fact, whenever there is a strong conflict between sources to be combined, the straightforward application of DS combination rule can produce a result in which certainty is assigned to the minority opinion [12].

A more refined approach is based on the Transferable Belief Model (TBM) proposed by Philips Smets in [13]. The TBM theory, like the Shafer formulation, relies on the concept of basic belief assignment function, but removes the assumption of $m(\emptyset) = 0$. This allows to omit the normalization constant in the Dempster’s rule of combination and conditioning.

**Definition 3 (Smets - operator $\otimes$):**

In the TBM, the combination rule is, therefore, defined in this way:

$$s_{ij} \triangleq s_i \otimes s_j = \{(m_i \otimes m_j)(\gamma_a); \gamma_a \in \Gamma\} \quad (4)$$

where:

$$m_{ij} \triangleq (m_i \otimes m_j)(\gamma_a) = \sum_{\gamma_b, \gamma_c \gamma_b \cap \gamma_c = \gamma_a} m_i(\gamma_b)m_j(\gamma_c). \quad (5)$$

Note that, for sake of clarity the following notation $m_i(\gamma_a) \otimes m_j(\gamma_a) \triangleq (m_i \otimes m_j)(\gamma_a)$ will be used indiscriminately in the rest of the paper.

The fact that $m(\emptyset) > 0$ can be explained in two ways: the open world assumption and the quantified conflict. The open world assumption reflects the idea that $\Omega$ might not be exhaustive, i.e. it might not contain all the possibilities. Under this interpretation, being $\emptyset$ the complement of $\Omega$, the mass $m(\emptyset) > 0$ represent the modelling errors, that is the fact that the truth might not be contained in $\Omega$. The second interpretation of $m(\emptyset) > 0$ is that there is some underlying conflict between the sources that are combined in order to produce the BBA $m$. Hence, the mass assigned to $m(\emptyset)$ represents the degree of conflict.

In particular, it can be computed as follows:

$$m_{ij}(\emptyset) = 1 - \sum_{\gamma_a \neq \emptyset \gamma_a \in \Gamma} m_{ij}(\gamma_a) \quad (6)$$

**III. An Introductory Example**

Let us consider a simple scenario, where three agents, whose network topology is depicted in Figure 1, observe the same event. Furthermore, let us assume that the objective of these agents is to perform a data aggregation in order to reach a common knowledge about such an event. To this end, let us suppose that agent 1 first collaborates with agent 2 and successively sets up a collaboration with agent 3. At this point, a question arises: what happens if agent 1 collaborates again with agent 2? Let us further investigate this situation.

![Fig. 1. Network topology.](image)

Every agent can apply the Smets combination rule to interact with its neighbors. When agent 1 collaborates with agent 2 they share the owned information to reach a new common knowledge ($s_{12}$). After that, when agent 1 sets up a collaboration with agent 3, they reach a new knowledge ($s_{123}$). Now if agent 1 and agent 2 collaborated again, by using the Smets combination rule, they would reach a new “wrong” knowledge ($s_{12123}$), where the information due to the first communication would be considered twice.

Therefore, in a distributed scenario a different combination strategy must be designed in order to overcome this issue. To this end, let us assume the current knowledge of an agent can be divided with respect to any of its neighbors in two parts, i.e. common knowledge and novelty. In particular, the common knowledge represents the portion of information shared by the agents, while the novelty is the novel portion of information brought by an agent. At this point, this issue can be simply overcome by restricting the aggregation among agents to the novelty, and then combining the obtained result with the common knowledge. In this way, when agent 1 and agent 2 collaborate again, they can reach a new knowledge avoiding to consider twice the result of their previous aggregation.
In the following, a proper formalization of this intuitive idea is proposed.

IV. PROBLEM SETTING

Let the network of agents be described by an undirected graph $G = \{V, E\}$, where $V = \{v_i : i = 1, \ldots, n\}$ is the set of nodes (agents) and $E = \{e_{ij} = (v_i, v_j)\}$ is the set of edges (connectivity) representing the point-to-point communication channel availability. A position $p_i \in \mathbb{R}^d$ in the $d^{th}$ dimensional space is associated to each node $v_i \in V$, with $i = 1, \ldots, n$.

In particular, an edge representing a connection between two agents exists if and only if the distance between these agents is less than or equal to their communication radius $r$, namely

$$E = \{e_{ij} : \|p_i - p_j\|_d \leq r, \ i \neq j\},$$

where $\| \cdot \|_d$ is the Euclidean norm in $\mathbb{R}^d$. Since the graph is undirect the existence of the edge $e_{ij}$ (from node $i$ to node $j$) implies the existence of the edge $e_{ji}$ (from node $j$ to node $i$). Therefore, in the following they will be used without distinction to indicate a connection between node $i$ and $j$. Moreover, we will refer to $N(i)$ as the neighborhood of agent $i$, namely the set of indices of the agents directly connected through an edge with agent $i$.

In the proposed framework a gossip algorithm [14] is defined as a triplet $\{S, \mathcal{R}, \epsilon\}$ where:

- $S = \{s_1, \ldots, s_n\}$ is the set containing the local state $s_i \in \mathbb{R}^q$ of each agent $i$ in the network.
- $\mathcal{R}$ is the local interaction rule ($\oplus$ binary operator) that, for any couple of agents $(i, j)$ with $e_{ij} \in E$, gives:
  $$\mathcal{R} : \mathbb{R}^q \times \mathbb{R}^q \rightarrow \mathbb{R}^q.$$
- $\epsilon$ is the edge selection process that specifies which edge $e_{ij} \in E(t)$ is selected at time $t$.

From an algorithmic point of view, a possible implementation of the gossip algorithm is given in Algorithm 1.

**Algorithm 1: Gossip Algorithm**

**Data:** $t = 0$, $s_i(0) \ \forall i = 1, \ldots, n$.
**Result:** $s_i(t_{stop}) \ \forall i = 1, \ldots, n$.

while stop condition do
  - Select an edge $e_{ij} \in E(t)$ according to $\epsilon$.
  - Update the states of the selected agents applying $\mathcal{R}$:
    $$s_i(t + 1) = s_i(t) \oplus s_j(t)$$
    $$s_j(t + 1) = s_j(t) \oplus s_i(t)$$
  - Let $t = t + 1$.
end

The following assumptions on the network of agents are made:

**Assumptions 1:**
- The network can be described by a connected undirected graph $G$.
- The communication range is limited by a maximum communication radius $r$.
- The communication among agents is asynchronous, gossip like [14].
- A distributed algorithm to build a spanning tree $T$ is available to the agents, for example by using [15], [16].
- Agents are capable to handle the storage of a proper set of data.

V. DISTRIBUTED DATA AGGREGATION VIA NETWORKED TBM

In this section, a local interaction rule $\mathcal{R}$ to perform the distributed TBM data aggregation over a network topology for a multi-agent system, under assumptions 1, is described. Let us assume $T = \{V, E\}$ to be the spanning-tree over the graph $G$ computed by the agents in a distributed fashion, where $V = \{v_i : i = 1, \ldots, n\}$ is the set of nodes (agents) and $E = \{e_{ij} = (v_i, v_j)\}$, $\mathcal{E} \subseteq E$ is a subset of the edges of $E$ required to build a spanning-tree.

**Definition 4 (S):**
Let $S(t) = \{s_1(t), \ldots, s_n(t)\}$ be the set of the agents states defined with respect to a finite frame of discernment $\Omega = \{\omega_1, \ldots, \omega_m\}$, where $s_1(t) = \{m_1(t, \gamma_1), \gamma_1 \in \Gamma\}$, $s_i(t) \in \mathbb{R}^{|\Gamma|}$ is the set of basic belief assignment (BBA) of agent $i$ over the power set $\Gamma(\Omega)$ at a given time $t \in \mathbb{N}$. Note that, in the following the time dependence will be omitted for sake of clarity if not strictly required.

Let us now introduce the binary operator $\otimes$, which is useful to break up any set of basic belief assignment with respect to any other one.

**Lemma 1:** Let us consider two sets of basic belief assignment (BBA) $s_k = \{m_k(\gamma_a) : \ \forall \gamma_a \in \Gamma\}$, and $s_i = \{m_i(\gamma_a) : \ \forall \gamma_a \in \Gamma\}$. It can be defined an operator $\otimes$:

$$s_k \triangleq s_k \otimes s_i = s_j$$

such that:

$$s_k = s_i \otimes s_j$$

In particular, each element of $s_j = \{m_j(\gamma_a) : \ \forall \gamma_a \in \Gamma\}$, can be computed recursively as follows:

$$m_k(\gamma_a) = \frac{\sum_{\gamma_b \supseteq \gamma_a} m_b(\gamma_a) m_i(\gamma_b)}{\sum_{\gamma_b \supseteq \gamma_a} m_i(\gamma_b)}$$

by starting from the element of the power-set with highest cardinality, $|\Gamma| = \{\Omega\}$, and moving down to the elements with cardinality equal to one, i.e., $\{\gamma_i \in \{\omega_i\}, \ i = 1, \ldots, n\}$, with $\gamma_1 = \emptyset$.

**Proof:** The proof is a simple consequence of the application of Smets operator $\otimes$. Let us assume $s_k$ can be
written as the Smets aggregation of $s_i$ and $s_j$:

$$s_k \triangleq s_i \otimes s_j = \left\{ (m_i \otimes m_j)(\gamma_i); \; \gamma_i \in \Gamma \right\}$$

where:

$$m_k(\gamma_a) \triangleq (m_i \otimes m_j)(\gamma_a)$$

$$= \sum_{\gamma_i \cap \gamma_c = \gamma_a} m_i(\gamma_b) \cdot m_j(\gamma_c)$$

$$= m_j(\gamma_a) \sum_{\gamma_b \subseteq \gamma_a} m_i(\gamma_b) + \sum_{\gamma_b \cap \gamma_c = \gamma_a} m_j(\gamma_b)m_i(\gamma_c).$$

At this point, by collecting with respect to $m_j(\gamma_a)$ the following expression is obtained:

$$m_j(\gamma_a) = \frac{m_k(\gamma_a) - \sum_{\gamma_b \cap \gamma_c = \gamma_a} m_j(\gamma_b)m_i(\gamma_c)}{\sum_{\gamma_b \subseteq \gamma_a} m_i(\gamma_b)}.$$ 

Therefore, $s_j = \left\{ m_j(\gamma_a); \; \forall \gamma_a \in \Gamma \right\}$ is obtained. 

**Definition 5 (R operator $\oplus$):**

Let $R$ be a rule to combine the basic belief assignments for two agents $(i, j)$ such that $e_{i,j} \in \hat{E}$ as follows:

$$s_i(t + 1) = s_i(t) \oplus s_j(t) = \left\{ \left( \hat{m}_i^j(t, \gamma_a) \otimes \hat{m}_j^i(t, \gamma_a) \right) \otimes \hat{m}_{i,j}(t, \gamma_a); \; \forall \gamma_a \in \Gamma \right\},$$

(10)

with $\otimes$ the Smets operator. Let us denote with $\hat{s}_i^j(t) = \left\{ \hat{m}_i^j(t, \gamma_a); \; \forall \gamma_a \in \Gamma \right\}$ the novelty of the agent $i$ with respect to the agent $j$, which can be computed recursively as follows:

$$\hat{m}_i^j(t, \gamma_a) = \frac{m_i(t, \gamma_a) - \sum_{\gamma_b \cap \gamma_c = \gamma_a} m_i(\gamma_b)m_{i,j}(t, \gamma_c)}{\sum_{\gamma_b \subseteq \gamma_a} m_{i,j}(t, \gamma_b)}$$

(11)

and $\hat{s}_{i,j}(t) = \left\{ \hat{m}_{i,j}(t, \gamma_a); \; \forall \gamma_a \in \Gamma \right\}$ (or equivalently $\hat{s}_{j,i}(t)$) is the common knowledge, i.e., the knowledge stored by both agents after their last aggregation, set to the neutral element $n = \{0, 0, \ldots, 0, 1\}$ of the TBM conjunctive rule before their first aggregation.

Note that, as a consequence of Lemma 1, for a given agent $i$ the following relation holds between the novelty and the common knowledge with any other agent $j$:

$$s_i(t) = \left\{ m_i(t, \gamma_a); \; \forall \gamma_a \in \Gamma \right\} = \hat{s}_i^j(t) \otimes \hat{s}_{i,j}(t)$$

(12)

Furthermore, for any couple of agents $(i, j)$, the related states $s_i$ and $s_j$ are equal if and only if they are completely described by their common knowledge, i.e., $s_i = s_j = \hat{s}_{i,j}$.

**Remark 1:** A few important remarks are now in order:

- In order to apply the local interaction rule $\mathcal{R}$, an agent must have stored all the most recent collaborations with its neighbors, that is $\left\{ s_i \oplus s_j; \; j \in \mathcal{N}(i) \right\}$.
- As only information concerning collaborations among (1-hop) neighbors are required, the algorithm is fully distributed and scalable in terms of memory requirement with respect to the size of the network.

At this point, in order to prove the convergence of the proposed algorithm, some properties concerning the local interaction rule $\mathcal{R}$ must be introduced.

**Lemma 2 ($\mathcal{R}$ properties):** The local interaction rule $\mathcal{R}$ defined according to eq. (10) has the following properties:

$$s_i \oplus s_j = s_j \oplus s_i$$

(commutativity)

$$s_i \oplus s_i = s_i$$

(idempotence)

$$(s_i \oplus s_j) \oplus s_k = s_i \oplus (s_j \oplus s_k)$$

(associativity)

(13)

for each triple $(i, j, k) : e_{i,j}, e_{j,k} \in \hat{E}$.

**Proof:** The properties can be proven by applying the definition given in eq. (10).

- **Commutativity:**
  Let us consider two agents $(i,j)$, then from Definition 5 we have:

$$s_i \oplus s_j = \left( \hat{s}_i^j \otimes \hat{s}_{i,j} \right) \otimes \hat{s}_{i,j}$$

$$= \left( \hat{s}_j^i \otimes \hat{s}_{i,j} \right) \otimes \hat{s}_{i,j}$$

$$= s_j \oplus s_i$$

where $\left( \hat{s}_i \otimes \hat{s}_j \right) = \left( \hat{s}_j \otimes \hat{s}_i \right)$ comes from the commutativity property of the Smets operator $\otimes$ and $\hat{s}_{i,j} = \hat{s}_{j,i}$ by definition.

- **Idempotence:**
  Let us suppose two agents $(i,j)$ at a given time $t$ have their BBA equal to their common knowledge (acquired at certain time previous $t$), that is $s_i = s_j = \hat{s}_{i,j}$, then we have:

$$s_i \oplus s_j = \left( \hat{s}_i^j \otimes \hat{s}_{i,j} \right) \otimes \hat{s}_{i,j}$$

$$= (n \otimes n) \otimes \hat{s}_{i,j}$$

$$= \hat{s}_{i,j}$$

$$= s_i$$

- **Associativity:**
  Let us consider a triplet of agents $(i, j, k)$ such that
Let us consider three agents \((i, j, k)\) such that \(e_{ij}, e_{jk} \in \hat{E}\) with their observations \(s_i(0), s_j(0), s_k(0)\) at time \(t = 0\). The following holds:

\[
s_i(0) \otimes s_j(0) \otimes s_k(0) = s_i(0) \oplus s_j(0) \oplus s_k(0) \tag{15}
\]

**Proof:** The lemma can be proven by applying the definition given in eq. (10) along with the properties given in eq. (12):

\[
s_i(0) \oplus s_j(0) \oplus s_k(0) = \left(s_i(0) \oplus s_j(0) \right) \oplus s_k(0) = \left(s_i(0) \otimes s_j(0) \otimes \bar{n}\right) \oplus s_k(0)
\]

where

\[
s_i(0) \otimes s_j(0) \otimes \bar{s}_{i,j}(0) = s_i(0) \otimes s_j(0) \otimes \bar{n}
\]

is due to the independence of the agents observation. Now, by defining \(s_z(0) = s_i(0) \otimes s_j(0)\), it follows that:

\[
s_z(0) \oplus s_k(0) = \left(s_z(0) \otimes \bar{s}_{z}(0) \otimes \bar{s}_{z,k}(0)\right) = s_z(0) \otimes s_k(0) \otimes \bar{n} = s_i(0) \otimes s_j(0) \otimes s_k(0)
\]

Let us now introduce the main result of the paper, that is the convergence of the proposed gossip algorithm towards the basic belief assignment (BBA) as in the centralized aggregation schema given in Definition 6.

**Theorem 1 (Distributed TBM):** Let us consider a gossip algorithm \(\{S, R, \epsilon\} \) over a spanning-tree \(\mathcal{T} = \{V, E\}\) with \(S\) and \(R\) defined respectively as in Definition 4 and Definition 5. Let us assume each agent \(i\) at time \(t = 0\) provides an independent set of observations described by the basic belief assignment \(s_i(0) = \{m_{i}(0, \gamma_a); \gamma_a \in \Gamma\}\). If \(\epsilon\) is such that \(\forall t \ni \exists \Delta t \in \mathbb{N}\) so that the time-variant forest \(\mathcal{F}(t, t + \Delta t)\) is connected, then there will exist a time \(t = \bar{t}\) so that:

\[
s_i(t') = s_1(0) \otimes s_2(0) \otimes \ldots \otimes s_n(0) \forall t' > \bar{t}, \tag{16}
\]

that is, each agent \(i\) converges toward the same BBA as in the centralized aggregation schema given in Definition 6.

**Proof:** The proof of the theorem consists of three steps. First, it will be proven that a steady-state exists for the proposed gossip algorithm. Successively, it will be proven that such an algorithm always converges toward a steady-state. Finally, it will be proven that the steady-state is unique and it is the same as the result of the centralized aggregation schema given in eq. (14).

1) **Steady-State Existence:** In order to prove the existence of a steady-state for the proposed gossip algorithm, it will be shown that a sufficient and necessary condition is that all the agents share the same state \(\bar{s}\). In fact, if all the agents have the same state \(\bar{s}\), according to the interaction rule given in eq. (10), they will always send the neutral element \(n\) for any further aggregation. Therefore, the state \(\bar{s}\) itself is a steady
state for the multi-agent system. Furthermore, let us prove by contradiction this condition to be necessary as well. To this end, let us consider a spanning-tree $T$ computed by the agents in a distributed fashion. Now, let us suppose two agents $i$ and $j$ have reached two different steady states over the network, that is $s_i(t) = s'$ and $s_j(t) = s''$. Therefore, according to the definition of a spanning-tree, there will always exist a (unique) path connecting the two nodes $i$ and $j$. Let us know consider for such a spanning-tree $T$ the path $p_{ij} = \{v_1, v_{k\in N_i}, \ldots, v_{h\in N_j}, v_j\}$ connecting these two agents $i$ and $j$. In particular, as agent $i$ has reached the state $s'$, its neighbor $k$ will always send to it the neutral element $n$ as novelty for any further aggregation. This implies that, the agent $k$ must have reached itself the same steady state $s'$ and be receiving the neutral element $n$ by its neighbors. The same argument can be applied to the agent $j$ and its neighbor $h$ with respect to the steady state $s''$. Now, by iterating this reasoning from both ends of the path there will be a cut where all the nodes on a side will have reached the same steady-state $s'$ as agent $i$, while on the other side all the agents will have reached the same steady-state $s''$ as agent $j$, as shown in Figure 3. Let us call $x$ and $y$ the two agents on the boundaries of the cut. Since, $x$ and $y$ have both reached a steady state, $s'$ and $s''$ respectively, they must be sending the neutral element $n$ as novelty to each other. However, the two steady states $s'$ and $s''$ have been supposed to be different, therefore the two agents $x$ and $y$ cannot be sending the neutral element $n$ to each other. Indeed, this would be possible only if the two steady states $s'$ and $s''$ were the same steady state $\bar{s}$, which gives the absurd. Therefore a steady state holds if and only if $s_i = s_j \quad \forall i, j \in N$.

2) Steady-State Convergence: In order to prove the convergence of the proposed algorithm towards a steady-state, let us consider a spanning tree $T$ computed by the agents in a distributed fashion over the network topology $\mathcal{G}$, as shown in Figure 2. Now, let us consider an interval of time $[t_0, t_0 + \Delta t_0]$ for which the forest $\mathcal{F}(t_0, t_0 + \Delta t_0)$ is connected. This implies that some agents play the role of leaves for the resulting spanning-tree $T$. According to the definition of the local interaction rule given in eq. (10), (at least) these agents will always send the neutral element $n$ to their fathers for any further aggregation (Figure 2-a). Now, let us consider a new interval of time $[t_1, t_1 + \Delta t_1]$ with $t_1 = t_0 + \Delta t_0 + 1$. We can use the same argument with respect to a new spanning-tree $T'$ obtained by removing the leaves from the original spanning-tree $T$. In fact, there are some other agents which play the role of leaves for the new spanning-tree $T'$ in the time interval $[t_1, t_1 + \Delta t_1]$. This implies again that (at least) these agents will always send the neutral element $n$ for any further aggregation to their fathers. At this point, since the number of agents is finite, by repeating this reasoning it will exist an interval $[t_h, t_h + \Delta t_h]$ after which the residual spanning tree $T^h$ will be composed of only one agent $i$, whose state $\bar{s}$ is the aggregation of all the observations available over the network (Figure 2-b). Let us now consider, a new spanning tree $T^{h+1}$ composed of such an agent $i$ and all of its one-hop neighbors. There will exist an interval $[t_{h+1}, t_{h+1} + \Delta t_{h+1}]$ after which the forest $\mathcal{F}(t_{h+1}, t_{h+1} + \Delta t_{h+1})$ is connected. As a result, all the agents belonging to this spanning tree will have reached the same knowledge as the agent $i$. This is due to the fact, that agent $i$ will be the only one to send a novelty different from $n$, and therefore any aggregation will let the other agents reach its state $\bar{s}$. By iterating the same reasoning, there will be an interval of time $[t_{2h}, t_{2h} + \Delta t_{2h}]$ for which the related spanning tree $T^{2h}$ will coincide with the original spanning tree $T$. At this point, all the agents will have reached the same state $\bar{s}$ as the agent $i$ (Figure 2-c). Therefore, according to the proof of existence, $\bar{s}$ is a steady state for the multi-agent system.

3) Steady-State Uniqueness: In order to prove the uniqueness of the steady state, it will be shown that any sequence of aggregations over the network, where each agent is considered at least once, is always the combination of the

![Figure 3. Novelty contraction over the spanning tree $T$ at different time-interval.](image-url)
Let us now provide a characterization of the convergence time with respect to a given edge selection process $\epsilon$.

**Lemma 4 (Convergence Time):** Let us consider an edge selection process $\epsilon$ such that $\forall t \exists \Delta t \in \mathbb{N}$, so that the forest $F(t, t + \Delta t)$ is connected. If $\exists M \in \mathbb{N}$ : $\Delta t < M \forall t$, then the convergence is reached by any agent at most at time $t = d \cdot M$, where $d$ is the diameter of the spanning tree $T$.

**Proof:** The proof of the lemma follows the same arguments of the steady-state convergence proof (V.2) by assuming that an upper bound is available to the time required for the forest to be connected. In particular, for sake of simplicity and with no lack of generality let us assume to start at time $t = 0$. Under this assumption, the information contraction process towards a single agent $i$ described in V.2 takes in the worst case, i.e., the leaves are the last agents to perform an aggregation, time $t_1 = (d/2) \cdot M$. In the same way, the information propagation process from such an agent $i$ to all the other agents over the network described in V.2 takes in the worst case, i.e., one of the leaves of the previous spanning-tree is the last agent to perform an aggregation, time $t_2 = (d/2) \cdot M$. Therefore, the overall time required to the algorithm to converge in the worst case scenario is $t_{\text{tot}} = t_1 + t_2 = d \cdot M$.

Thus proving the theorem. Details concerning the algorithm execution are provided in Appendix I.

### Appendix I

**Example of Distributed Data Aggregation**

In the following, an example of distributed data aggregation involving a system of 5 agents (sources) observing an event is given. Two facts are assumed to be possible for the observed event, hence the following frame of discernment is defined $\Omega = \{a, b\}$ and the following power-set is obtained. $\Gamma = \{\emptyset, a, b, \{a, b\}\}$. Observations collected by the agents are summarized in Table I.

<table>
<thead>
<tr>
<th>Set $\emptyset$</th>
<th>Agent 1</th>
<th>Agent 2</th>
<th>Agent 3</th>
<th>Agent 4</th>
<th>Agent 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\emptyset$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>${a}$</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
<td>0.2</td>
<td>0.3</td>
</tr>
<tr>
<td>${b}$</td>
<td>0.8</td>
<td>0.7</td>
<td>0.8</td>
<td>0.7</td>
<td>0.4</td>
</tr>
<tr>
<td>${a, b}$</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.3</td>
</tr>
</tbody>
</table>

**TABLE I**

Observations collected by the system of 5 agents.

In the case the centralized TBM aggregation schema is considered, according to Definition 6 the result of the data aggregation is shown in Table II.

<table>
<thead>
<tr>
<th>Set $\emptyset$</th>
<th>Agent 12</th>
<th>Agent 123</th>
<th>Agent 1234</th>
<th>Agent 12345</th>
<th>C.TBM</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\emptyset$</td>
<td>0.23</td>
<td>0.341</td>
<td>0.471</td>
<td>0.63499</td>
<td>0.63499</td>
</tr>
<tr>
<td>${a}$</td>
<td>0.05</td>
<td>0.011</td>
<td>0.0035</td>
<td>0.00213</td>
<td>0.00213</td>
</tr>
<tr>
<td>${b}$</td>
<td>0.71</td>
<td>0.647</td>
<td>0.5183</td>
<td>0.36285</td>
<td>0.36285</td>
</tr>
<tr>
<td>${a, b}$</td>
<td>0.01</td>
<td>0.001</td>
<td>0.0001</td>
<td>0.00003</td>
<td>0.00003</td>
</tr>
</tbody>
</table>

**TABLE II**

Centralized TBM: final result and progressive aggregation.

Fig. 4. Multi-Agent system: solid (black) lines represent the network topology $G$, dashed (red) lines describe the spanning tree $T$.

Let us now consider the distributed TBM aggregation schema proposed in this work. In particular, Fig 4 depicts the multi-agent system where the solid (black) lines describes the network topology $G$, while the dashed (red) lines represents the spanning tree $T$ computed in a distributed fashion by the agents. The edge selection process for the proposed example is described in Table III. For sake of simplicity, only collaborations among agents which augment their common knowledge, i.e., the novelty is not the neutral element $n$ at least for one of the two agents, have been considered. Each agents has a task where the common knowledge with its neighbors is stored. In particular, by assuming each mass can be represent with a double (4 bytes in our representation), the memory occupancy for each agent is equal suitable for robotic applications where the network topology is likely to switch over time.

**VI. Conclusions**

In this paper the data aggregation problem for a multi-agent system has been investigated. Agents are assumed to be independent reliable sources which collect data and collaborate to reach a common belief. A distributed protocol for distributed data aggregation which was proved to converge to the basic belief assignment (BBA) given by a centralized aggregation based on the Transferable Belief Model (TBM) conjunctive rule has been provided.

Future work will be mainly focus on two directions. First, we would like to perform a real implementation of the proposed data-fusion protocol over a sensor network. Second, we would like to relax the requirement of a spanning tree. Indeed, this would make the approach more robust and...
to $4 \times 2 \times 8 = 64$ bytes, where 2 is the maximum number of neighbors in the spanning-tree $T$ and 8 is the cardinality of the power-set $\Gamma$. In the proposed example, the system of agents reaches a converge toward the same BBA after 9 steps. In particular, it can be noticed that each single agent has the same BBA (Tables X, XI, XII) as in the case of the centralized aggregation schema based on the TBM conjunctive rule (Table II).

<table>
<thead>
<tr>
<th>Time</th>
<th>Agent 1</th>
<th>Agent 2</th>
<th>Agent 3</th>
<th>Agent 4</th>
<th>Agent 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t=1$</td>
<td>$s_1$</td>
<td>$s_2$</td>
<td>$s_3$</td>
<td>$s_4$</td>
<td>$s_5$</td>
</tr>
<tr>
<td>$0$</td>
<td>$0.4781$</td>
<td>$0.4781$</td>
<td>$0.4781$</td>
<td>$0.4781$</td>
<td>$0.4781$</td>
</tr>
<tr>
<td>$(a)$</td>
<td>$0.00213$</td>
<td>$0.00213$</td>
<td>$0.00213$</td>
<td>$0.00213$</td>
<td>$0.00213$</td>
</tr>
<tr>
<td>$(b)$</td>
<td>$0.36285$</td>
<td>$0.36285$</td>
<td>$0.36285$</td>
<td>$0.36285$</td>
<td>$0.36285$</td>
</tr>
</tbody>
</table>

### TABLE III

**EDGE SELECTION PROCESS.**

<table>
<thead>
<tr>
<th>Set #</th>
<th>Agent 1</th>
<th>Agent 2</th>
<th>Agent 3</th>
<th>$s_1(t) \oplus s_2(t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\emptyset$</td>
<td>$0$</td>
<td>$0$</td>
<td>$0$</td>
<td>$0.23$</td>
</tr>
<tr>
<td>$(a)$</td>
<td>$0.1$</td>
<td>$0.1$</td>
<td>$0.1$</td>
<td>$0.05$</td>
</tr>
<tr>
<td>$(b)$</td>
<td>$0.8$</td>
<td>$0.8$</td>
<td>$0.8$</td>
<td>$0.71$</td>
</tr>
<tr>
<td>$(a,b)$</td>
<td>$0.1$</td>
<td>$0.1$</td>
<td>$0.1$</td>
<td>$0.01$</td>
</tr>
</tbody>
</table>

### TABLE IV

**DISTRIBUTED TBM: $T=1$, $s_1 \oplus s_2$.**

<table>
<thead>
<tr>
<th>Set #</th>
<th>Agent 1</th>
<th>Agent 2</th>
<th>Agent 3</th>
<th>$s_1(t) \oplus s_2(t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\emptyset$</td>
<td>$0.23$</td>
<td>$0.23$</td>
<td>$0$</td>
<td>$0.341$</td>
</tr>
<tr>
<td>$(a)$</td>
<td>$0.05$</td>
<td>$0.05$</td>
<td>$0.1$</td>
<td>$0.01$</td>
</tr>
<tr>
<td>$(b)$</td>
<td>$0.71$</td>
<td>$0.71$</td>
<td>$0.8$</td>
<td>$0.647$</td>
</tr>
<tr>
<td>$(a,b)$</td>
<td>$0.01$</td>
<td>$0.01$</td>
<td>$0.1$</td>
<td>$0.001$</td>
</tr>
</tbody>
</table>

### TABLE V

**DISTRIBUTED TBM: $T=2$, $s_1 \oplus s_3$.**

<table>
<thead>
<tr>
<th>Set #</th>
<th>Agent 2</th>
<th>Agent 4</th>
<th>$s_2(t) \oplus s_4(t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\emptyset$</td>
<td>$0.23$</td>
<td>$0.23$</td>
<td>$0$</td>
</tr>
<tr>
<td>$(a)$</td>
<td>$0.05$</td>
<td>$0.05$</td>
<td>$0.2$</td>
</tr>
<tr>
<td>$(b)$</td>
<td>$0.71$</td>
<td>$0.71$</td>
<td>$0.7$</td>
</tr>
<tr>
<td>$(a,b)$</td>
<td>$0.01$</td>
<td>$0.01$</td>
<td>$0.1$</td>
</tr>
</tbody>
</table>

### TABLE VI

**DISTRIBUTED TBM: $T=3$, $s_2 \oplus s_4$.**

<table>
<thead>
<tr>
<th>Set #</th>
<th>Agent 3</th>
<th>Agent 5</th>
<th>$s_3(t) \oplus s_5(t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\emptyset$</td>
<td>$0.341$</td>
<td>$0$</td>
<td>$0.5395$</td>
</tr>
<tr>
<td>$(a)$</td>
<td>$0.011$</td>
<td>$0.1$</td>
<td>$0.0609$</td>
</tr>
<tr>
<td>$(b)$</td>
<td>$0.647$</td>
<td>$0.4$</td>
<td>$0.4533$</td>
</tr>
<tr>
<td>$(a,b)$</td>
<td>$0.001$</td>
<td>$0.001$</td>
<td>$0.00003$</td>
</tr>
</tbody>
</table>

### TABLE VII

**DISTRIBUTED TBM: $T=4$, $s_3 \oplus s_5$.**

<table>
<thead>
<tr>
<th>Set #</th>
<th>Agent 1</th>
<th>Agent 2</th>
<th>$s_1(t) \oplus s_2(t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\emptyset$</td>
<td>$0.423$</td>
<td>$0$</td>
<td>$0$</td>
</tr>
<tr>
<td>$(a)$</td>
<td>$0.011$</td>
<td>$0.05$</td>
<td>$0.017$</td>
</tr>
<tr>
<td>$(b)$</td>
<td>$0.647$</td>
<td>$0.8$</td>
<td>$0.575$</td>
</tr>
<tr>
<td>$(a,b)$</td>
<td>$0.001$</td>
<td>$0.001$</td>
<td>$0.00003$</td>
</tr>
</tbody>
</table>

### TABLE VIII

**DISTRIBUTED TBM: $T=5$, $s_1 \oplus s_2$.**

<table>
<thead>
<tr>
<th>Set #</th>
<th>Agent 3</th>
<th>Agent 4</th>
<th>$s_3(t) \oplus s_4(t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\emptyset$</td>
<td>$0.341$</td>
<td>$0.5395$</td>
<td>$0.5395$</td>
</tr>
<tr>
<td>$(a)$</td>
<td>$0.003$</td>
<td>$0.011$</td>
<td>$0.0069$</td>
</tr>
<tr>
<td>$(b)$</td>
<td>$0.5183$</td>
<td>$0.433$</td>
<td>$0.3825$</td>
</tr>
<tr>
<td>$(a,b)$</td>
<td>$0.00001$</td>
<td>$0.00003$</td>
<td>$0.00003$</td>
</tr>
</tbody>
</table>

### TABLE IX

**DISTRIBUTED TBM: $T=6$, $s_1 \oplus s_3$.**

<table>
<thead>
<tr>
<th>Set #</th>
<th>Agent 5</th>
<th>$s_5(t) \oplus s_5(t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\emptyset$</td>
<td>$0.63499$</td>
<td>$0.4781$</td>
</tr>
<tr>
<td>$(a)$</td>
<td>$0.00213$</td>
<td>$0.00213$</td>
</tr>
<tr>
<td>$(b)$</td>
<td>$0.36285$</td>
<td>$0.36285$</td>
</tr>
<tr>
<td>$(a,b)$</td>
<td>$0.00003$</td>
<td>$0.00003$</td>
</tr>
</tbody>
</table>

### REFERENCES


