



HAL
open science

Analysis of temporal consistency management protocols for resource-constrained multi-agent systems

Mohamed Limame, Julien Henriet, Christophe Lang, Nicolas Marilleau

► To cite this version:

Mohamed Limame, Julien Henriet, Christophe Lang, Nicolas Marilleau. Analysis of temporal consistency management protocols for resource-constrained multi-agent systems. *Engineering and Applied Science Research*, 2022, 49 (1), pp.65-72. hal-03692831

HAL Id: hal-03692831

<https://hal.science/hal-03692831v1>

Submitted on 10 Jun 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0
International License

1 **KKU International Engineering Conference 2021 (KKU IENC2021)**

2 **May 12-14, 2021 - Khon Kaen, Thailand**

3 **Analysis of temporal consistency management protocols for resource-constrained multi-**
4 **agent systems**

5 M. Limame*, J. Henriet**, C. Lang** and N. Marilleau*

6 * UMMISCO, IRD (Institut of Research for the Development), Sorbonne University, Bondy,
7 93143, France.

8 **FEMTO-ST Institute, Franche-Comté University, Besançon, CNRS, 25000, France.

9 Contact Email address: mohamed.limame@etu.sorbonne-universite.fr

10

11

12 **Abstract**

13 Multi-Agent Systems (MAS) are distributed system composed of a set of connected nodes, that
14 communicate and share data. Since nodes do not have access to a central clock, the consistency
15 of shared data is one of the important issues raised by this type of system. Approaches to
16 establish temporal consistency of these shared data have been designed and implemented in
17 the past. This paper presents and analyses existing protocols for managing temporal
18 consistency in order to identify those that are in line with the specificities of weakly adjoining
19 MAS with limited resources and connectivity. We propose a classification and a comparative
20 analysis of these protocols for use within this category of MAS.

21 **Keywords:** Consistency management, temporal consistency, shared data, synchronization, multi-
22 agent systems, multi-agent systems with limited resources and connectivity.

23

24 **1. Introduction**

25 In Multi-Agent Systems (MAS), time is of much importance in the interpretation of
26 collected and exchanged knowledge, and in the determination of the agents' behavior,
27 especially during data exchanges. In this article, we are interested in approaches derived from
28 distributed systems ensuring temporal consistency of shared data. Thus, we propose a
29 classification and a comparative analysis of protocols from the literature on distributed systems
30 that can be used in MAS with limited resources and connectivity.

31 According to [1], a MAS is a distributed system composed of autonomous entities (called
32 agents). They share knowledge and interact with each other to reach a collective goal beyond
33 the scope of a single agent. Such approach could be applied to model various sensitive

34 applications as environmental monitoring. However, it raises many scientific challenges such
35 as temporal consistency management. Indeed, data separately collected, stored as distributed
36 agent knowledges must be consistent to safe their gathering and to build a collective
37 knowledge. We limit our study to constrained MAS that are composed of mobile agent
38 qualified with low computing resources, data storage, energy and connectivity.

39 According to [2], a distributed system is a set of autonomous computers connected by a
40 computer network that appears to its users as a single coherent system. These computers share
41 information and resources over a wide geographical area, which in some applications has given
42 rise to problems of temporal consistency, particularly in real-time applications used in
43 factories, aeronautics, space vehicles or military applications: more broadly in version
44 management and control of concurrent access in distributed database systems.

45 In distributed systems, several approaches have been identified to address this issue. It is
46 imperative to understand the characteristics of the agents in order to distinguish among the
47 approaches used in distributed systems those that are in adequacy with their needs and
48 constraints.

49 The main contribution of this article is a classification of the most used synchronization
50 approaches and algorithms in distributed systems taking into account the specificities of multi-
51 agent systems. This classification work will help researchers in the choice of the most
52 appropriate synchronization technique to meet their MAS limited resources and connectivity
53 constraints.

54 In what follows, we will define consistency and introduce the temporal consistency. We will
55 then present synchronization approaches that we consider interesting to establish temporal
56 consistency within a MAS. Then we define the classification criteria before presenting the
57 qualification of the selected approaches according to these criteria. A comparative analysis will
58 synthesize the whole of our study.

59

60 **2. Consistency : concept and methods**

61 In this part, we describe what the consistency is and we give some ways in order to maintain
62 this consistency.

63 *2.1 Consistency definitions*

64 There are several definitions of consistency in the literature as confirmed by [3]. [4] defines
65 consistency as a means of measuring the absence of the concept of negative affectivity which

66 is similar to neurosis. In linguistics, according to [5, 6], a discourse (text or dialogue) can be
67 qualified as coherent when its parts "go together" and thus structure the discourse. [7, 8] brings
68 an epistemological vision by defining consistency as a cognitive theory translating a level of
69 constraints satisfaction contained in individual knowledge in its different forms: explanatory,
70 analogical, deductive, perceptual and conceptual.

71 In distributed systems, according to [9] the concept of consistency can be seen as the
72 conservation of an explanatory or conceptual logic ensured by an algorithm and constraints. Its
73 management is essential to avoid calculation errors on the one hand and anomalies of algorithm
74 execution on the other hand.

75 In distributed systems, consistency is usually global and associated with the whole system.
76 In multi-agent systems, for their part, according to [10], the system's consistency is defined by
77 the compatibility of collected and stored information by each agent. It is, therefore, defined at
78 the individual agent level. In this case, MAS consistency is an emergent consistency that results
79 from the individual consistencies.

80 [11] introduces temporal consistency in a distributed system by the original concept of "time
81 window". Indeed, the notion of temporal consistency that is applied to a set of information
82 aims at ensuring that the information belonging to this set has been produced in the same time
83 window. According to [12], two types of temporal consistency can be distinguished in
84 distributed systems with respect to the accuracy requirement level of the time notion. Applied
85 to SMAs, it is possible to define a strict temporal consistency in which all the elements and
86 knowledge held by all the agents have exactly the same time stamp and a non-strict temporal
87 consistency in which differences in the stamping of elements and knowledge may exist between
88 agents. Temporal consistency at the global level is obtained when the temporal consistency of
89 each element of the multi-agent system is achieved.

90 *2.2 Methods*

91 Among all the approaches that can be applied to establish temporal consistency, there
92 is the approach that uses a notion of time synchronization between all agents through the help
93 of a synchronization protocol on the one hand, and the correction approach applied by pairs of
94 agents following the use of a conflict resolution technique on the other hand. In this case, one
95 of the two involved agents re-orders the values of the shared data.

96 It is possible to distinguish different tools and methods for solving conflicts related to
97 the consistency of shared data. According to [13] arbitration is a means used by MAS to solve

98 conflicts and leads to the definition of behavioral rules that act as constraints on all agents. [13]
99 also proposes the elaboration of a priority agreement. The definition of priority rules avoids
100 the appearance of conflicts. However, in case of conflict, the system will use the unequal weight
101 of the agents. It is also possible to establish a voting and election procedure. According to [14],
102 votes in MAS are methods allowing each agent to express its preferences among possible
103 decisions. Finally, the negotiation is a communication process to reach a mutually accepted
104 agreement according to [15].

105 Different synchronization protocols are based on physical and logical clocks. A
106 physical clock is a physical process coupled with a time measurement method. Most physical
107 clocks are based on cyclic processes based on an oscillator and a counter. A logical clock is a
108 time-stamping method that records the chronological and causal relationships in a distributed
109 system to establish events global classification of the system actions and processes.

110 *2.2.1 Physical clock synchronization algorithms*

111 The concept of synchronization is based on a physical clock source whose data is
112 propagated in a network. The NTP protocol, Cristian's and Berkeley's algorithms are based on
113 physical clocks.

114 **The NTP protocol** uses a semi-stratified time sources hierarchical system based on
115 UTC (Coordinated Universal Time) as time reference. As described in [16] each level of this
116 hierarchy is called a "stratum" and is assigned a number. A server synchronized on a server of
117 stratum n runs at stratum $n+1$.

118 **Cristian's algorithm** [17] is a clock based algorithm used to synchronize the local
119 clock of a distributed system element with a remote external time server. A system element
120 sends a request to the time server in order to receive the updated time.

121 **The Berkeley algorithm** is described in [18]. Unlike Cristian's algorithm, the time
122 server is active. It periodically interrogates each element of the distributed system on its local
123 clock. On the basis of the responses, it calculates an average time and asks all elements of the
124 system in order to advance or slow down their clocks to the new calculated clock.

125 The implementation of the NTP protocol in a MAS requires the setting up of an inter-
126 agent connection respecting the NTP architecture to send time data and requires continuous
127 connectivity of the stratum $n^{\circ}1$ agents to a reference time (Coordinated Universal Time). For
128 a use of Berkley and Cristian algorithms in a MAS, it is necessary to define one agent as the

129 server agent and to ensure continuous connectivity of this agent with all the other agents for
130 the synchronization process.

131 2.2.2 *Logic clock synchronization algorithms*

132 The concept of a logical clock was initially introduced in [19]. Several authors [20, 21,
133 22] have then proposed the use of logical clocks to detect the causal precedence relationship
134 between events.

135 **Lamport's clock** [19] shows that synchronization does not necessarily have to be
136 absolute and that it can be deduced from relationships between events. The proposed
137 scheduling allows to assign a logical clock or stamp to all events in a distributed system.

138 With **Mattern's clock** [20], each component p of the distributed system has an integer
139 vector called *stamp* in which each component $stamp[i]$ is the estimation by p of the Lamport
140 clock value of process i .

141 With **Matrix clocks** [23], each process p of a distributed system of n processes has an
142 $n \times n$ matrix of *stamps* in which each component $stamp[i]$ is the estimation by p of the Mattern
143 clock value of process i .

144 For an implementation of these approaches in a MAS, it is necessary to set up an
145 exchange between agents according to the described principles of exchange between processes.
146 The stamp(s) must be sent during inter-agent interactions to ensure synchronization.

147 2.2.3 *Clock synchronization in wireless sensor networks*

148 Various applications using wireless sensor networks, especially in monitoring and data
149 fusion, require that all the nodes have synchronized clocks. However, the synchronization
150 methods traditionally used in industry are not necessarily suitable for a use in sensor networks
151 due to problems related to energy consumption. For example, the NTP protocol, although
152 widely used for clock synchronization over the Internet, is not suitable for a use in wireless
153 networks because it is too power-hungry. New approaches adapted for a use in this type of
154 networks have thus emerged in recent years.

155 **Reference Broadcast Synchronization Protocol** (RBS) is based on a receiver-to-
156 receiver synchronization scheme. As described in [24], each node synchronizes its local clock
157 with all the other clocks of the nodes within its transmission range. The nodes in the transmitter
158 transmission range record the local clock that receives its message. Then the receivers
159 exchange the registered reception clocks with each other. Thus, each receiver can estimate the
160 offset of its clock by eliminating transmission latencies.

161 **Time Synchronization Protocol for Sensor Networks** (TPSN) is based on a
162 transmitter-receiver synchronization scheme. The algorithm described by [25] works in two
163 steps. In the first step, a hierarchical structure is established in the network, assigning each
164 node a level in a synchronization tree. Then comes a second synchronization step during which
165 the i level nodes are synchronized with the $i-1$ level nodes pair by pair.

166 **Flood Time Synchronization Protocol** (FTSP) [26] implements a synchronization in
167 which the root node periodically transmits a unique synchronization message to the nodes that
168 are within transmission range. From the timestamp value at transmission and the one at
169 reception, a node can determine its clock deviation from the root node. If a node does not
170 receive synchronization messages for a certain period of time, it declares itself the new root.

171 **Delay measurement time synchronization for wireless sensor networks** (DMTS) is
172 based on transmitter-receiver synchronization in which the sender and several receivers are
173 synchronized at the same time. In this protocol [27], a master node is chosen as time server and
174 broadcasts its clock. All receivers within range measure the transmission delay and set their
175 local clock by assigning to it the received master clock plus the estimated transfer delay.

176 **Consensus Time Synchronization** (CCS) technique aims at reduce clock gaps between
177 nearby nodes and converges all nodes to a common gap. The main idea of this algorithm [28]
178 is to compensate over several iterations the clock gaps between the nodes of the system using
179 the average time synchronization algorithm described in [29]. The nodes broadcast their local
180 clock in the network so that each node can estimate the rate of clock deviation. Then, the nodes
181 broadcast their estimated rate of virtual clock deviation, allowing the receiver nodes to combine
182 this with their relative deviation estimates to adjust their own virtual clock. Thus, all nodes
183 then converge asymptotically to the same clock.

184 **Gradient Time Synchronization Protocol** (GTSP) was introduced in [30]. In this
185 protocol, network nodes periodically broadcast a synchronization beacon with their
186 neighboring nodes. Using a simple update algorithm, they agree on a common logical clock
187 with their neighbors. There is no reference node. Clock synchronization is done by pairs of
188 nodes, both parties must agree on a common logical clock frequency and an absolute value of
189 the logical clock. By applying this approach, the logical clock of each node converges to a
190 common logical clock.

191 **Reachback Firefly Algorithm** (RFA) is inspired by synchronization in large biological
192 swarms where individuals follow simple coordination strategies. The canonical example is the

193 synchrony of fireflies observed in parts of Southeast Asia. The behavior of these systems can
194 be modeled as an array of pulse-coupled oscillators where each node is an oscillator that
195 periodically emits a self-generated pulse. . An example of modeling is presented in [31]. By
196 observing the pulses of the other oscillators, a node slightly adjusts the phase of its own
197 oscillator. This simple feedback process allows nodes to closely align their phases and achieve
198 synchronization.

199 Since wireless sensor networks can be considered as particular MAS, for the
200 implementation of these protocols in a MAS, we just need to apply the same processing
201 performed by wireless sensor nodes to the MAS agents level.

202 *2.2.4 Synchronization using data*

203 **Synchronization of multi-agent systems based on data evolution (SMASDEV) [32]**
204 relies on the content of an event and its data evolution to sort events. The aim of this new
205 approach is to establish overall consistency at the level of a MAS by enabling each agent to re-
206 establish a chronological order of the data it receives from other agents, based on its knowledge
207 and without using a clock (physical or logical). In SMASDEV, each agent records in memory
208 its personal perception of the data evolution that it has collected with a certain frequency in
209 relation to his local clock. The agent can thus predict the data future evolution. Thus, the agent
210 must have a model of evolution in adequacy with the monitored data. In order to move from a
211 personal perception to a global perception, each agent of the system must first ask the other
212 agents to transmit it the collected data and then place them in relation to its personal perception.
213 In this way, the agent will can identify the positioning of all received data in relation to his
214 clock. As a result, the agents will be in phase with the data evolution model over time.

215 **Optimistic Pilgrim Protocol [33]** is a consistency management protocol of share data
216 in a distributed system with replicated data. It works on the basis of a token circulating on a
217 unidirectional logical ring and containing a data structure with updates of the shared data. These
218 data are transported by the token called *Pilgrim*. As it passes over one node, the Pilgrim brings
219 the updates from the other nodes and retrieves the values that have been locally modified to
220 disseminate them through the the cooperating sites. In this protocol, only the owner of a shared
221 data can deposit the modifications of this data on the token. For the implementation of this
222 approach in an SMA, the agents represent the nodes and the connectivity must be continuous
223 between the agents and the token resource, otherwise the synchronization cannot take place.

224 3. Methods classification with evaluation criteria

225 3.1 Criteria definition

226 For an effective relevance measurement of each synchronization approach, the chosen
227 classification criteria must sufficiently characterizing constraints in a MAS with limited
228 resources and connectivity. We distinguish two families of criteria.

229 *The "synchronization details" criteria family* qualifies the synchronization approaches
230 and the obtained synchronization level. It concerns the synchronization principle, the type of
231 consistency, the synchronization object, the type of synchronization and the fault tolerance.

232 *The "energy" criteria family* allows to qualify the synchronization approaches needs
233 especially in terms of memory and power. This family concerns the performed calculation, the
234 volume exchanged, the number of required messages to synchronize data between 2 agents and
235 the number of required messages to synchronize data between N agents. These indicators allow
236 to estimate the required energy by the synchronization approach.

237 3.1.1 The synchronization details

238 In this family, the **synchronization principle** specifies the synchronization mechanism
239 used by the approach.

240 The **consistency type** allows to know the synchronization level in the network between
241 agents. Among consistency's types found in literature [34-35], we consider the four following
242 types:

243 *The atomic consistency level* reflects a strict global synchronization degree among all
244 the agents in the network. At the end of a synchronization cycle, when each agent has received
245 the shared data, all agents in the network must have the same system state. Synchronization at
246 the atomic level is quite complex to deploy in a distributed system.

247 *The sequential consistency level* reflects L. Lamport's vision of synchronization and
248 focuses on the events order, not the occurrence time of the event. At the end of a
249 synchronization cycle, when each agent has received the shared data, all the agents on the
250 network must present the same shared events order. This is therefore a less strict degree of
251 synchronization than the atomic level.

252 *The causal consistency level* is based on the causal relationship between the events
253 occurring in the network of agents. At the opposite to the sequential consistency level, at the
254 end of a synchronization cycle, when each agent has received the shared data, all the agents in
255 the network do not necessarily have the same events order because the events synchronization

256 is only based on the events linked by a causal relationship. The remaining events are therefore
257 not known by all the agents.

258 *The released coherence level* reflects the lowest degree of synchronization compared to
259 those mentioned above.

260 **The synchronization object** specifies the nature of the synchronized element(s). The
261 obtained consistency through synchronization is a temporal consistency if the synchronized
262 element is a "time" data.

263 **The synchronization type** qualifies the synchronization process in relation to its
264 triggering: whether it is systematic or non-systematic.

265 **The fault tolerance criterion** indicates whether the synchronization approach supports
266 the presence of one or more failures in the agent fleet.

267 3.1.2 *The "energy" family of criteria*

268 **The realized calculation criterion** characterizes the operations needed to launch the
269 synchronization approach.

270 **The exchanged volume** characterizes the required volume exchanged between two
271 agents to ensure the synchronization.

272 **The number of required messages to synchronize a data item between 2 agents** is
273 the number of exchanged messages required to ensure the synchronization between two agents.

274 **The number of required messages to synchronize data between N agents** is the
275 number of exchanged messages required in order to ensure the synchronization between N
276 agents. The number of exchanged messages allows to quantify the agent energy consumption
277 of the execution of the synchronization approach.

278 3.2 *Classification of the studied synchronization approaches*

279 Our classification is presented in the Table 1 below. Five of the fifteen approaches
280 considered in this paper support fault tolerance by default. Even if the rest of the approaches
281 do not have this feature, it can be deployed at the time of their implementation.

282 For the studied synchronization approaches based on clocks or data exchange, the
283 network topology does not usually constitute an obstacle for these approaches . Indeed, except
284 for the NTP and TPSN approaches which need a hierarchical structure, the agents in the rest of
285 studied approaches must be able to communicate with each other independently of the
286 topology.

287

288 **Table1** – Classification of the studied synchronization approaches

289

290 *The values in the column "type of consistency" are presented with 'A' for atomic consistency, 'S' for*
 291 *sequential consistency, 'C' for causal consistency and 'R' for relaxed consistency. The criterion*
 292 *"synchronization type" is represented by the column "Systematic synch.". The value 'X' is filled in only*
 293 *if the consistency is systematic. In the same way for the column "Fault Toler.", the value 'X' is only filled*
 294 *in if the fault tolerance is supported by the synchronization approach. In the column «Realized*
 295 *Calculation", 'op' designates operation and 'n' refers to the agents' number in the MAS.*

296

The synchronization's details criteria's family						The « energy » criteria's family			
Synchr. Principle	Consist. type	Synchr. Object	Systematic Synch.	Fault Toler.	Realized calculation	Exchanged volume	Nbr msg to synchr. 2 agents	number of messages to synchr. N agents	
Physical clock synchronization algorithms									
NTP	sharing one stamp	R	stamp	x	x	op subtraction + op division	stamp	2	$n-m+1$ n receiver level, m transmitter level
Cristian's Algorithm	through a server time	S	clock	x		op addition + op division	Round-trip + server clock time	2	$2n$
Berkeley's Algorithm	through a server time	R	Offset	x	x	op addition + op division	server : n clocks	3	$3n$
Logic clock synchronization algorithms									
Lamport clocks	sharing one stamp	c	Data + stamp	x		op comparison + op increment	local stamp and synch	1	$n-1$
Mattern clocks	share a vector of stamps	S	vector of stamps	x		$n-1$ op comparison + op increment stamp	vector local stamps and synch	1	$n-1$
Matrix clocks	share a matrix. Stamps	S	Matrix of stamps	x		$n-1$ op comparison + op increment stamp	matrix local stamps and synch	1	$n-1$
Clock synchronization in wireless sensor networks									
RBS	clock sharing	R	transmitter: data; receiver: clock			Add. + div. + offset calcul.	Shared data + k clocks (k neighbors)	2	$1+(n-1)*k$; k: nbr neighboring nodes to (n-1) nodes
TPSN	sharing one stamp	R	Synchronization packet.			Tree-treeing + new clock calculation	a stamp	2	$n-m+1$ stamp with n receiver level and m transmitter level
FTSP	clock sharing	R	ID + root stamp + data	x	x	Offset calculation	stamp + shared data	2	$1+k$: 1 msg transmitter + k msg receivers to neighborhood
DMTS	sharing one stamp	R	Root stamp			calculation new clock	stamp	1	1 msg from sender to receiver

CCS	sharing one stamp	R	stamp and deviation rate	x	x	calculation rate of deviation	stamp + rate of deviation	2 * number of iterations	2*(n-1) * number of iterations
GTSP	clock sharing	R	synch beacon and clock freq.	x	x	update clock frequency + calculate. New clock	clock + clock frequency	2	2*(n-1)
RFA	sharing a state	S	state vector	x		state vector and objective fct calculate	Data synchr. Status vector	1	1 msg from sender to receivers
Synchronization using data									
Smasdev	data assimilation	C	Data with an evolution model			data manip. & simul. Of evol.	Shared data tuple	1	n-1
Optimistic pilgrim	Res. 1 reserve. & distrib. V. token	A	Shared Data			jeton manipulation	Shared Data	6	3n+1

297

298 All the studied approaches, based on physical or logical clocks, ensure synchronization
 299 between two elements by sending one to three messages. In the approaches used in wireless
 300 sensor networks, a synchronization between two nodes can be achieved with one or two
 301 messages.

302 We find that for the RFA and DMTS approaches, the number of sent messages needed
 303 for the MAS global synchronization does not vary by increasing or decreasing the number of
 304 agents. In fact, the sender sends a single broadcast message to all the agents.

305 In wireless sensor networks, the studied approaches require a limited amount of
 306 memory space in order to achieve temporal consistency.

307 The storage requirement for data-based approaches depends on the synchronization
 308 object size and the result of the function used for synchronization.

309 Among the studied approaches, only the Optimistic Pilgrim allows atomic consistency.

310 Nine of the fifteen studied approaches, independently of their fields of application,
 311 allow to perform data synchronization in addition to the time notion of synchronization (stamp,
 312 clock, ...).

313 All of the studied synchronization approaches based on physical and logical clock
 314 exchanges systematically trigger on the occurrence of events. Whereas some of the studied
 315 approaches on wireless sensor networks such as RBS, TPSN and DMTS aren't systematically
 316 launched when events occur.

317

318 **4. Results and discussion**

319 This classification lets one determine which synchronization technique is best suited to
320 his/her application constraints in MAS with limited resources and connectivity. In the case of
321 agents with limited resources, their computing capacity may also be limited, memory space
322 may be reduced, and message exchanges may be restricted. The more the synchronization
323 approach needs are reduced, the longer the approach can be applied. This point can be crucial,
324 especially in cases of application on routes and paths, such as during mapping or monitoring
325 missions. This interpretation is detailed in the following section.

326 Concerning memory space, the synchronization approach based on matrix clocks is
327 characterized by a large memory space requirement. Indeed, the storage of all the matrices
328 associated with each event (local or received) is costly. Similarly, with a lesser degree, the
329 synchronization using Mattern clocks has a relatively expensive memory space requirement.
330 The use of these approaches can have a negative impact on the robustness of the system. On
331 the other hand, DMTS, NTP, TPSN and Cristian's algorithm are characterized by a relatively
332 low memory space requirement. Their need consists essentially in storing, at the receiving
333 agent level, the clock or time stamp to be synchronized.

334 The Lamport's clock, the Berkeley Algorithm, RBS, FTSP, CCS and the Optimistic
335 Pilgrim protocol require a small computing capacity reduced to the power needed to compute
336 time averaging, the offset or writing on a token. The Cristian and Berkeley algorithms are two
337 approaches adapted to MAS in which it is possible to manage time centrally.

338 In terms of energy consumption, it is the CCS synchronization approach that requires
339 the largest number of messages, since messages are sent to all the agents in the network over
340 several iterations until shared data gaps are eliminated. This can be a direct factor of agent
341 energy exhaustion. Moreover, the CCS synchronization type is systematic, making it costly in
342 terms of energy consumption.

343 Similarly, the NTP, FTSP, Cristian and Berkeley algorithms being executed in a regular
344 and periodic way, are likely to consume a lot of energy if the launch frequency is important.
345 Concerning the studied approaches that are launched at the event occurrence (RFA or Mattern's
346 clocks for example), the energy consumption increases slightly at each launch, which is suitable
347 for the use in a multi-agent system with limited resources and connectivity.

348 Energy consumption for SMASDEV approach for which synchronization is not
349 systematic, remains relatively low and increases proportionally with the complexity of the
350 shared data evolution model and the distribution density of the agent network.

351 Although the seven studied approaches used in wireless sensor networks can be applied
352 in a MAS, it would be preferable to limit as much as possible the frequency of their execution
353 in an environment in which agents have limited resources and connectivity.

354 Based on Lamport's approach, the comparison of clock values does not allow to deduce
355 a causal relationship between two a and b events. Indeed, if $H(a) < H(b)$ does not necessarily
356 mean that $a \rightarrow b$. Moreover, in case of a high traffic, messages may not be received. With these
357 limitations, we propose not to use it in a MAS with limited resources and connectivity.

358 The choice of synchronization approach can also be determined according to the nature
359 of the shared data. For example, if the shared data only concerns the notion of time (stamp,
360 time matrix, ...), the choice can be one of the TPSN, DMTS, CCS or GTSP approaches designed
361 to establish temporal synchronization and in adequacy with the studied MAS (limited
362 resources and connectivity). On the other hand, if the shared data does not concern the notion
363 of time, we propose the use of the RFA, Smasdev, RBS or FTSP approaches which are in
364 adequacy with the studied MAS and which are sorted according to the order of degree of
365 coherence from the strictest to the least strict. We discard from this list the Optimistic Pilgrim
366 requiring that all agents can communicate continuously and therefore not very compatible with
367 an MAS made up of weakly accounting agents.

368 Protocols that require the least resources have to be privileged for MAS with limited
369 resources and connectivity. Beyond these considerations, the synchronization approach to use
370 depends strongly on the synchronization context and the required level of synchronization.

371

372 **5. Conclusions**

373 This article presents a study of different possible solutions to the problem of temporal
374 synchronization of shared data between agents among a MAS with limited resources and
375 connectivity. We have presented and compared several approaches used in distributed systems
376 to solve the lack of a centralized clock. We analyzed them for the use in a MAS with limited
377 resources and connectivity. The approach has consisted in starting from existing solutions in
378 the universe of distributed systems in order to identify those which are in adequacy with the
379 constraints of the studied multi-agent system, in particular the limitation of resources.

380 In order to carry out this work, we first identified evaluation criteria that characterize
381 the various constraints of a MAS in terms of resources, architecture but also quality of
382 synchronization. Then, we classified the identified synchronization approaches according to
383 these criteria to end with an analysis of the obtained results. In addition to the used criteria to
384 qualify the resource requirements of an approach and to answer the question of the suitability
385 of the approach to the studied multi-agent system, the rest of the evaluation criteria provide a
386 precise assessment of the synchronization approach quality.

387 In the light of this study, we now plan to test and compare performances of the
388 approaches that appeared most relevant and adapted to this type of MAS. Thus, we plan to
389 develop a multi-agent simulator for different application areas such as territory surveillance
390 using a fleet of UAVs.

391

392 **6. Acknowledgements**

393 We are grateful to all of those with whom we have had the pleasure to work during this and
394 other related projects.

395

396 **7. References**

- 397 [1] CHAIB-DRAA, Brahim. Industrial applications of distributed AI. *Communications of the ACM*,
398 1995, vol. 38, no 11, p. 49-53.
- 399 [2] VAN STEEN, Maarten et TANENBAUM, Andrew S. *Distributed systems*. Leiden, The
400 Netherlands : Maarten van Steen, 2017.
- 401 [3] OLSSON, Martin, HANSSON, Kjell, LUNDBLAD, Ann-Marie, *et al.* Sense of coherence:
402 definition and explanation. *International Journal of Social Welfare*, 2006, vol. 15, no 3, p. 219-229.
- 403 [4] STRÜMPFER, D. J. W., GOUWS, J. F., et VIVIERS, M. R. Antonovsky's Sense of Coherence
404 Scale related to negative and positive affectivity. *European Journal of Personality*, 1998, vol. 12, no 6,
405 p. 457-480.
- 406 [5] GROSZ, Barbara et SIDNER, Candace L. Attention, intentions, and the structure of
407 discourse. *Computational linguistics*, 1986.
- 408 [6] CARBAUGH, Donal. *Conversational Coherence: Form, Structure, and Strategy*. 1989.
- 409 [7] THAGARD, Paul. *Coherence in thought and action*. MIT press, 2000.
- 410 [8] THAGARD, Paul. *Hot thought: Mechanisms and applications of emotional cognition*. MIT press,
411 2008.
- 412 [9] SABLATNIG, Sven Grottko Andreas Kopke Jan et WOLISZ, Jiehua Chen Ruedi Seiler Adam.
413 Consistency in Distributed Systems. 2007.

**KKU International Engineering Conference 2021 (KKU-IENC2021)
May 12-14, 2021 - Khon Kaen, Thailand**

- 414 [10] VU, Quang Anh Nguyen, GAUDOU, Benoit, CANAL, Richard, *et al.* Coherence and robustness
415 in a disturbed MAS. In : *2009 IEEE-RIVF International Conference on Computing and Communication*
416 *Technologies*. Ieee, 2009. p. 1-4.
- 417 [11] THOMESSE, Jean Pierre. The fieldbuses. *IFAC Proceedings Volumes*, 1997, vol. 30, no 7, p.
418 13-23.
- 419 [12] CAUFFRIEZ, Laurent et DEFRENNE, Jean. An Experimental Platform for the Reliability-
420 Availability Evaluation of Intelligent Distributed Systems. *IFAC Proceedings Volumes*, 1997, vol. 30, no
421 7, p. 545-550.
- 422 [13] FERBER, Jacques et WEISS, Gerhard. *Multi-agent systems: an introduction to distributed*
423 *artificial intelligence*. Reading : Addison-Wesley, 1999.
- 424 [14] LANG, Jérôme. Logical preference representation and combinatorial vote. *Annals of*
425 *Mathematics and Artificial Intelligence*, 2004, vol. 42, no 1, p. 37-71.
- 426 [15] BUSSMANN, Stefan et MULLER, Jorg. A negotiation framework for co-operating
427 agents. *Proceedings of CKBS-SIG*, 1992, p. 1-17.
- 428 [16] MINAR, Nelson. A Survey of the NTP Network. 1999.
- 429 [17] CRISTIAN, Flaviu. Probabilistic clock synchronization. *Distributed computing*, 1989, vol. 3, no
430 3, p. 146-158.
- 431 [18] GUSELLA, Riccardo et ZATTI, Stefano. The accuracy of the clock synchronization achieved by
432 TEMPO in Berkeley UNIX 4.3 BSD. *IEEE transactions on Software Engineering*, 1989, vol. 15, no 7, p.
433 847-853.
- 434 [19] LAMPORT, Leslie. Time, clocks, and the ordering of events in a distributed system. In
435 : *Concurrency: the Works of Leslie Lamport*. 2019. p. 179-196.
- 436 [20] MATTERN, Friedemann, *et al.* *Virtual time and global states of distributed systems*. Univ.,
437 Department of Computer Science, 1988.
- 438 [21] FIDGE, Colin J. Timestamps in message-passing systems that preserve the partial ordering.
439 1987.
- 440 [22] SCHWARZ, Reinhard et MATTERN, Friedemann. Detecting causal relationships in distributed
441 computations: In search of the holy grail. *Distributed computing*, 1994, vol. 7, no 3, p. 149-174.
- 442 [23] RAYNAL, Michel et SINGHAL, Mukesh. Logical time: Capturing causality in distributed
443 systems. *Computer*, 1996, vol. 29, no 2, p. 49-56.
- 444 [24] ELSON, Jeremy, GIROD, Lewis, et ESTRIN, Deborah. Fine-grained network time
445 synchronization using reference broadcasts. *ACM SIGOPS Operating Systems Review*, 2002, vol. 36,
446 no SI, p. 147-163.
- 447 [25] GANERIWAL, Saurabh, KUMAR, Ram, et SRIVASTAVA, Mani B. Timing-sync protocol for
448 sensor networks. In : *Proceedings of the 1st international conference on Embedded networked sensor*
449 *systems*. 2003. p. 138-149.

**KKU International Engineering Conference 2021 (KKU-IENC2021)
May 12-14, 2021 - Khon Kaen, Thailand**

- 450 [26] MARÓTI, Miklós, KUSY, Branislav, SIMON, Gyula, *et al.* The flooding time synchronization
451 protocol. In : *Proceedings of the 2nd international conference on Embedded networked sensor systems.*
452 2004. p. 39-49.
- 453 [27] PING, Su. Delay measurement time synchronization for wireless sensor networks. *Intel*
454 *Research Berkeley Lab*, 2003, vol. 6, p. 1-10.
- 455 [28] MAGGS, Michael Kevin, O'KEEFE, Steven G., et THIEL, David Victor. Consensus clock
456 synchronization for wireless sensor networks. *IEEE sensors Journal*, 2012, vol. 12, no 6, p. 2269-2277.
- 457 [29] SCHENATO, Luca et FIORENTIN, Federico. Average TimeSynch: A consensus-based
458 protocol for clock synchronization in wireless sensor networks. *Automatica*, 2011, vol. 47, no 9, p. 1878-
459 1886.
- 460 [30] SOMMER, Philipp et WATTENHOFER, Roger. Gradient clock synchronization in wireless
461 sensor networks. In : *2009 International Conference on Information Processing in Sensor Networks.*
462 IEEE, 2009. p. 37-48.
- 463 [31] RAMÍREZ-ÁVILA, Gonzalo Marcelo, KURTHS, Jürgen, DEPICKERE, Stéphanie, *et*
464 *al.* Modeling fireflies synchronization. In : *A Mathematical Modeling Approach from Nonlinear Dynamics*
465 *to Complex Systems.* Springer, Cham, 2019. p. 131-156.
- 466 [32] LIMAME, Mohamed, HENRIET, Julien, LANG, Christophe, *et al.* Synchronisation d'horloge
467 dans un système multi-agents. In : *APIA.* 2019.
- 468 [33] GARCIA, Eric, GUYENNET, Hervé, HENRIET, Julien, *et al.* Towards an optimistic
469 management of concurrency: a probabilistic study of the pilgrim protocol. In : *International Conference*
470 *on Computer Supported Cooperative Work in Design.* Springer, Berlin, Heidelberg, 2005. p. 51-60.
- 471 [34] BRZEZINSKI, Jerzy, SOBANIEC, Cezary, et WAWRZY尼亚K, Dariusz. From Session Causality
472 to Causal Consistency. In : *PDP.* 2004. p. 152-158.
- 473 [35] RAYNAL, Michel et SCHIPER, André. *A suite of formal definitions for consistency criteria in*
474 *distributed shared memories.* IRISA, 1995.

475

476 **8. Reviewer**

477 No preference